LIBRARY OF THE
UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN

510.84
Il6r
no.613-617
cop.2

[redacted]

Digitized by the Internet Archive
in 2013

http://archive.org/details/platolessononinf615ches

510.84
Il6r
no.615
cop.2
Math

UIUCDCS-R-74-615

A PLATO LESSON ON INFORMATION STRUCTURES

by

Michel Jean Chesnel

January, 1974

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

THE LIBRARY OF THE
FEB 15 1974
UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN

UIUCDCS-R-74-615

A PLATO LESSON ON INFORMATION STRUCTURES

by

Michel Jean Chesnel

January, 1974

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

This work was supported in part by the Department of Computer Science, University of Illinois at Urbana-Champaign, and was submitted in partial fulfillment for the Master of Science degree in Computer Science, 1974.

ACKNOWLEDGMENT

The author wishes to express special gratitude to his thesis advisor Professor Edward M. Reingold for his supervision and helpful discussions during the research for this thesis. Special thanks also goes to Professor Marshall Mickunas, Alan Davis, and Ervin Wilson whose students cooperated in making valuable comments.

The helpful work of Ms. Lynn Waverly and Ms. Sharon Phillips in preparing this manuscript is gratefully acknowledged. Finally, I give my thanks to my parents, for their support and continuous faith in me, without which I know I could not have even started this thesis.

TABLE OF CONTENTS

Page

1. INTRODUCTION---------- 1

1.1 Nature of the Problem---------- 1

1.2 Computer-Based Education---------- 1

1.2.1 Drill and Practice---------- 3
1.2.2 Tutorial Teaching Mode---------- 4
1.2.3 Inquiry Teaching Mode---------- 5

1.3 The PLATO IV System---------- 5

2. DEFINITION OF A STRATEGY---------- 7

2.1 Four Basic Principles---------- 7

2.2 Frame Contents and Presentation---------- 8

2.3 Choice of a Mode---------- 9

2.4 Branching---------- 11

3. THE LESSON---------- 12

3.1 Purpose and Scope---------- 12

3.2 Description---------- 12

4. EVALUATION OF THE LESSON---------- 15

4.1 Choice of the Students---------- 15

4.2 Procedure---------- 16

4.3 Students' Comments---------- 17

4.4 Conclusion---------- 19

LIST OF REFERENCES---------- 20

APPENDIX---------- 21

1. INTRODUCTION

1.1 Nature of the Problem

The purpose of the study was to write a lesson in the TUTOR language for the PLATO IV computer assisted instruction system of the University of Illinois. The lesson STR1 written by the author deals with information structures and will be one of the PLATO lessons for the introductory computer science courses.

To provide background, a brief description of computer assisted instruction is given, followed by a description of the PLATO IV system.

1.2 Computer-Based Education

Computer-based education is a flexible approach to educational problems. The traditional approach, i.e., the teacher and printing press, made mass education possible. Nowadays, with the greater need of mass and individualized education, another approach is necessary. Computer-based education could be the answer to this problem.

For more than a decade now numerous efforts have been made to use computer technology to get a highly individualized education. Various computer-based education systems are in use throughout the United States, 85 such systems are listed in Helen A. Lekan's Index to Computer Assisted Instruction (1970).

The main characteristic of a computer assisted instruction system is, of course, the use of a computer. This computer provides two significant capabilities: memory and logic. The memory feature is also provided by other teaching aids, such as books, films, and tapes, but no aid provides the memory of students' responses in a form which is directly useful for automatic processing. Furthermore, only computer assisted instruction takes care of the communication problem on an individualized basis; only computer assisted instruction is adaptative and these last characteristics exist because of the powerful logic of the computer used in each computer assisted instruction system.

Any computer assisted instruction system can be divided into two main parts: the hardware and the software. The computer is only one component of the hardware, generally the less visible to the student. Each student interacts with the computer via a console. An interface is generally needed between the computer and the consoles. A console typically consists of a graphic display device, such as a television monitor display, a C.R.T. or a plasma display panel, and a keyset. The student or the author can read in the computer through the graphic display, and the keyset is used to make requests and responses for the student and to compose and edit lessons for the author. Other devices can be available such as a light pen, an image projector, an audio-tape for playback and recording capability.

The software consists of the normal software which is implemented on every computer running in time sharing, so that more than one console can be connected to the computer and operate simultaneously. Each student has the impression that he is the only individual who is interacting with the system.

Any computer assisted instruction system must have at least three features: 1) computer assisted instruction must be able to evaluate a student's response by comparing it to various answers predicted by the teacher or author (this evaluation must occur immediately after the student's answer), 2) computer assisted instruction must provide the teacher or the author with detailed data on student performance, 3) computer assisted instruction must make creation and modification of lessons easy for the author.

The principal aim of a computer assisted instruction system is to achieve an efficient individualization of instruction. The way of a computer assisted instruction adjusts a learning environment according to students' responses and requests can be seen as a "mode." There are three main student modes: drill and practice, the tutorial teaching mode, and the inquiry teaching mode. An ideal computer assisted instruction system would combine all of these student modes.

1.2.1 Drill and Practice

Drill and practice consists of presenting to the student a set of small exercises or problems to solve without

having previously shown him the material with which he is supposed to know how to solve these drills. In this mode the student does not learn much new material; he is simply checking how well he knows the material. The computer assisted instruction system prints questions to the student, reads his answer, evaluates its validity, returns appropriate messages, and loads in its memory all the necessary data reflecting the performance of the student. In this mode the student is involved rather intensively in the use of the console; nevertheless he needs to know very little about the way the computer assisted instruction system works, since he uses his natural language (English).

1.2.2 Tutorial Teaching Mode

Tutorial teaching is a second mode in which the teacher (or the author) and the computer assisted instruction system take responsibility for the student. This mode was the only one used on computer assisted instruction systems for a long time. It consists of two main parts: a demonstration when the student learns concepts and a supervised practice session when the student has to demonstrate his understanding of the concepts which have been previously presented to him. In this mode the lesson is generally linear so that all the students go through a very similar path. However, "help" and "branch" sequences are provided so that the instruction is adapted to each individual. Since the student has relatively little freedom in this

the author must be very familiar with both the language of the computer and the philosophy of teaching. The sequencing in this mode is essential and it is much more complex than frame writing because the problem of sequencing is closely tied to subject matter knowledge and to teaching experience.

1.2.3 Inquiry Teaching Mode

The inquiry mode is a third type of computer assisted instruction application. In this mode the student asks questions to the system by using his keyset. Because the student determines himself what he needs and can work entirely at his own rate, the education is really personalized. Another advantage of this mode is that the student needs not know anything about the computer assisted instruction system since communication between the system and the student is again through natural language. The author has to implement routines which locate the required information. If the language of the system is appropriate, this task is not too time-consuming.

1.3 The PLATO IV System

The PLATO IV system (Programmed Logic for Automatic Teaching Operation) uses a high-speed digital computer, a Control Data 6500, as the logic of the whole system. An N.I.U. (Network Interface Unit) is the link between site controllers and the computer. Each site controller can handle up to 32 consoles or terminals.

Each terminal is composed of a plasma display panel, a keyboard, a slide projector, a touch panel, an audio device (only few terminal currently have a touch panel and an audio device). The plasma display contains 512 digitally addressable positions along each axis and has an inherent memory. Thus, once the information is on the screen, no rewriting has to be done as it was the case in the PLATO IV system that used a television display. The computer provides the terminal only with new points to be shown and points to be erased. The actual design of the PLATO IV system makes possible the connection of about 4,000 student terminals within 150 miles of the University of Illinois.

The system is running under a special operating system, including time sharing for the 4,000 terminals and files and memory management to handle the huge amount of various data used by all the students and authors. A special teaching-oriented language, TUTOR, has been implemented on the system. Another important feature of the PLATO IV system is the collection of complete record of students' responses and behavior through lessons.

2. DEFINITION OF A STRATEGY

Whatever the mode used, there are some common basic principles which apply to lesson writing on a computer assisted instruction system. A survey of these principles will be presented in the following lines. Even before starting to write a lesson, one has to know what sort of students the lesson is intended for. Obviously, if the students are obliged to take the lesson the task of the author will be much more difficult than if the students are "volunteers" and highly motivated. In the latter case, even if the lesson is boring the students will be willing to go through it anyway because the material is attractive to them. The same kind of remark will apply if the students are high school students or college students. For each of these cases the same lesson will have to be written differently.

Unfortunately, no systematic rule exists and only a highly experienced teacher can really know the different ways of writing a lesson according to the students who are going to take it.

2.1 Four Basic Principles

It is difficult to state specific rules for lesson writing. However, when one has to write a lesson one must respect the four basis principles.

First of all is the principle of small steps, which consists of programming a large number of small and easy-to-take steps so that a student can proceed from knowing very little about a subject to its mastery by going through such a lesson. The second principle is the principle of active responding. The student not only reads explanations and looks at the examples, but also works out some problems and drills. The lesson must use most of the facilities of the computer assisted instruction system to be interactive. The student must have the impression that he has something in front of him with which he can have a "dialog." The third principle is the principle of immediate confirmation. With this principle the student can confirm his answer immediately. The last principle is the principle of self-pacing. Some people naturally learn more rapidly or more slowly than others, and each student will learn better if he can go at his own pace.

2.2 Frame Contents and Presentation

On the PLATO IV system the material is presented to the student on a screen. The contents of such a screen is called the "frame" or "page." Whenever the frame is full the screen is erased and, generally, the next frame or page is displayed. According to the principle of small steps, the frame is composed of small paragraphs and/or graphics. Each paragraph contains an idea or concept. The presentation of each new idea or concept is a difficult task for the author. In order to be efficient, one must always remember some techniques which make the

presentation agreeable to the student and easy to recall too. The text must be easy to understand and without any ambiguities. Graphics and drawings must be used as often as possible. Variety is another way to increase the interest of the student and to avoid boredom. The teacher must be aware that the student cannot remember everything at once, so whenever it is possible and reasonable, redundancy has to be introduced into the lesson. The meaningfulness of the text is another very important variable in regard to learning and retention of knowledge. The more meaningful the material is to the student, the easier it is to learn and recall. The readability is also an important variable in regard to learning. There are three main ways to increase readability: 1) use short and easy words, 2) use short sentences, and 3) use as few technical words as possible. Finally, the student must be aware of what he will see, so extensive use of topic heading must show the main points being presented.

2.3 Choice of a Mode

For the lesson STR1 the author has made the assumption that the college students who will go through this lesson can be considered mature and highly motivated. In spite of this assumption, the author has tried to write a lesson which is interesting for less motivated students. Based upon this assumption that the students are highly motivated, the author has chosen to write the lesson in the inquiry mode which has been described briefly in the first part of this thesis.

The lesson contains various topics and for each topic there is a corresponding question or set of questions. The student is free to go through these topics in any order he wants. However, if he does not know anything about the lesson he is advised to follow the "natural" path which is provided. To do so he will have to press the NEXT key at the end of each paragraph and frame, and the lesson will become a linear sequence of topics.

Another option is available to the student. He can choose between the three following "modes": 1) text and questions, 2) text only, or 3) questions only. The first mode is useful to the students who go through the lesson for the first time. The second mode is mostly for the students who go through the lesson for the first time, but who do not think it is worthwhile to go through questions and drills. The third mode is devoted to the students who already know the material. The flexible structure of the lesson allows the students who are only interested in one or more parts of the lesson to see them without going through all of the other parts. This is precisely a characteristic of the inquiry mode.

The students have to know the contents of the lesson or at least to know how to get to a page which contains the contents of the lesson. So the structure of the lesson has to be such that the students can know at any time what the main parts of the lesson are. This is done by implementing a table of contents accessible via the TERM key. From this page the

students are able to branch to any important part of the lesson by typing the corresponding key(s). Finally, the structure of the lesson must be clear not only to the author but above all to the students.

2.4 Branching

Whatever option the students might choose, the NEXT, BACK, and TERM keys have a special use. By pressing the NEXT key, the student will go to the next paragraph, graphic, or drawing within a frame. At the end of a frame the NEXT key leads the students to the next frame. The BACK key restarts the frame when pressed within a frame. When the student is at the end of a frame, the BACK key takes him to the previous one. The TERM key associated with a code (up to eight characters) branches from any place in the lesson to the unit associated with the code. The students have to know these codes, so a good policy is to implement a page which is accessed via TERM "term" where all codes and associated material are displayed.

Some other keys are also distinguishable, namely the DATA, LAB, HELP, and ERASE keys. DATA and LAB keys allow the author to implement more branching than only the two regular branching NEXT and BACK. The HELP key is generally used to give some hints to the students when they are confused. Finally, the ERASE key enables the students to erase part of their answer or whatever they write on the terminal (the shift-erase erases all the characters written by the students).

3. THE LESSON

3.1 Purpose and Scope

The lesson was written to give some basic knowledge about information structures. The principal topics covered in this lesson are the following: linear lists, i.e., stacks, queues, and deques; and linked and sequential storage allocation. These concepts are introduced to the student and examples are provided to illustrate these concepts. For stacks, queues, and deques, precise graphic representations are shown on the screen and the student can insert and delete items to acquire a better understanding of how these structures can be modified. At the end of each topic, a set of questions on the material presented is available.

As it was said before, the lesson is intended for college students beginning their studies in computer science. The lesson is self-sufficient, i.e., no particular prior knowledge is required and terminology is introduced where appropriate.

3.2 Description

The lesson can be divided into five parts: how to use the lesson, what the lesson is all about, introduction to information structures, linear lists (stacks, queues, deques), and storage allocation. Drills are associated with the last three parts to allow the student to see how well he has assimilated the material he has just gone through.

1) How to use the lesson - This section explains the three "modes" the student can select: a) see all the text and drills, b) only text, or c) only drills. This section also explains how to use the NEXT and BACK keys and what their purposes are. Finally, in this part the "term" options are explained.

2) What the lesson is all about - This part introduces all the topics which will be presented later on in the lesson. In this section is discussed why information structures are important in computer science and what one can do with these structures: how to manipulate, modify, create, and delete them, and what the basic omponent of a structure is.

3) Introduction to information structures - In this part the concept of information structures is actually introduced to the students. An example is used to ease the understanding. The representation of a deck of cards is used to show a node of a structure and its different fields. With the same example one can see how to access the information of any node through links. Information structures are not static so a basic algorithm to add a new node is shown. Two drills are associated with this part. First, a set of questions dealing with the accessing of field of several nodes. Secondly, the students are asked to type the two basic steps to insert a new node to a very simple information structure.

4) Linear lists - At that point the students are supposed to know what an information structure is in general, so

in this part a special kind of information structure, linear lists, is introduced. Then the various sorts of linear lists are described: stacks, queues, and deques. For each kind of linear list definition and properties are shown, then a graphic representation is displayed on the screen. Finally, the students can insert and delete elements to these lists. This part is ended by a set of five questions randomly chosen from twelve questions referring to the basic properties of these linear lists.

5) Storage allocation - Up to that point no precise mention was made about the actual implementation of all the data or information within the memory. Two basic techniques to implement information structures are described: sequential allocation where data are in successive locations, and linked allocation where one field of each node is reserved to point at the next node so that consecutive nodes do not have to be physically in contiguous memory locations. A basic technique for handling linked allocation is also discussed: the use of a list of available space. In this part the student is asked to formulate the logical steps to insert and delete a node in a stack and in a queue.

4. EVALUATION OF THE LESSON

When developing computer assisted instruction, the following question occurs frequently: how can the effectiveness of a teaching program be measured? Most educators believe that the basic purpose of learning is to change behavior. If the student does not do anything differently at the end of the learning experience than he did at the beginning, what has been the value of the program?

As previously stated, the purpose of the present study was, in part, to write a lesson in TUTOR language for the PLATO IV computer assisted instruction system and to evaluate the effectiveness of this lesson. The experiment which will be described was designed to accomplish this goal. Two groups of students were given the same test to evaluate how well the lesson objectives were attained by the students and to provide a relative measure of the effectiveness of this teaching method.

4.1 Choice of the Students

The students who participated in the present experiment were members of a University of Illinois computer science class (CS 201, Professor Mickunas). The students were undergraduates (except for one graduate student), mostly of junior or senior standing:

- one freshman
- one sophomore
- eight juniors
- six seniors
- one undergraduate irregular
- one graduate

The class had no unusual characteristics and most of the students had no previous experience with automatic programmed instruction. Finally, both groups of students participating in the experiment had no prior knowledge of the material covered by the lesson. The fact that these students had no prior knowledge of the material presented in the lesson was the main reason why they were selected for this experiment.

4.2 Procedure

No pre-test was given since no students were familiar with the material presented in the lesson. Upon completion of the instruction sequence, both groups were given a test, so that the test scores could be used as a variable for analysis. The questions used were to test how well the objectives of the lesson material had been attained. The test contained fourteen questions. The test was scored on the basis of the total number of correct responses. The average score was 11.3 or 81%. An outline of the score appears on page 17.

OUTLINE OF THE SCORE

Test Score	Percentage	Number of Students	Percentage
6	43%	1	5.6%
9	64%	2	11.1%
10	71%	4	22.2%
11	79%	2	11.1%
12	86%	2	11.1%
13	93%	4	22.2%

One point should be taken into account when considering the information presented above. The test seemed to be insufficiently difficult to provide a full range of scores; the lowest score was 6 (43%), and the test limited several students, i.e., three scores of 14 or 100% were obtained. The students participating in the experiment began by receiving an introduction to the PLATO IV system and becoming familiar with the student terminal and keyset. Then each group attended one session of the lesson. The longest completion time was about twice the shortest, giving a ratio of about 2 to 1. The shortest time was 45 minutes.

4.3 Students' Comments

After completing the lesson the students were asked various questions, and to write a critical evaluation of the PLATO IV system as well as the lesson.

Some students' attitudes toward media may not be a direct measure of their educational values. However, it appears that student attitudes that are related to how well they felt they learned are good measures of effectiveness. The students were asked whether they "enjoyed using the lesson to learn this material." One student disagreed, three were uncertain, twelve agreed, and two strongly agreed (no one selected "strongly disagreed"). The overall attitude of the students toward the use of PLATO IV was highly favorable: 82% of the students were favorable to PLATO IV, only 18% were uncertain, none were unfavorable to the use of the PLATO IV system.

Most of the dislikes expressed by the students concerned technical problems associated with the teaching program. They expressed dissatisfaction with some fragments of the lesson, almost exclusively with some lengthy drills. However, some students noticed that they were worthwhile. Students also felt the need to ask specific questions about the material which was often not contained in the "help" sequences.

As it was discussed previously, most of the students were not familiar with the PLATO IV system. Some students expressed that they had various difficulties in reviewing parts of the lesson or in skipping to another part of the lesson. Such facilities were provided mainly via the NEXT, BACK, and TERM keys and explained at the beginning of the lesson. However, some students had difficulties using these facilities.

This may be due to a certain lack of efficient explanation and/or due to the fact that most of the students were not familiar enough with the PLATO IV system and were inclined to use only the NEXT key. In effect, the introductory explanation of the PLATO IV system was very brief and it seemed that more time should have been devoted to this part.

4.4 Conclusion

From personal communication with some students who took the lesson and from the results of the test and the evaluation, it seems that computer assisted instruction and particularly the lesson written by the author can be used in computer science for introductory courses.

To summarize, the present study has shown that the lesson material can be taught on a computer assisted instruction system such as the PLATO IV system. However, the students felt a need for more flexibility in the lesson, so future work in this area should involve better designed and more extensive "help" sequences, references, and facilities to branch anywhere in the lesson.

LIST OF REFERENCES

Bitzer, D. L. and Johnson, R. L. "PLATO: A Computer-based System Used in the Engineering of Education." Proceedings of the Institute of Electrical and Electronic Engineers, 59 (June, 1971), 960-68.

Bitzer, D. L. and Skaperdas, D. "PLATO IV: An Economically Viable Large Scale Computer-based Education System." Paper presented at the National Electronics Conference, Dec., 1968.

Fry, Edward B. Teaching Machines and Programming Instruction. New York: McGraw-Hill, 1963.

Knuth, Donald E. The Art of Computer Programming. Vol. 1. Reading: Addison-Wesley, 1969.

Meredith, J. C. "Machine as Tutor," in The Educational Technology Reviews Series. Englewood Cliffs, New Jersey: Educational Technology Publications, 1972.

Smith, R. G., Jr. Innovations in Teaching and Training. Detroit: American Data Processing, Inc., 1968.

Stolurow, Lawrence. Computer Assisted Instruction. Detroit: American Data Processing, Inc., 1968.

TUTOR User's Memo. "Introduction to TUTOR: PLATO IV Version." University of Illinois, January, 1973.

APPENDIX

The following documentation of the source code of STR1 and STR2 follows the sequential construction of the lesson. Each unit is discussed individually, and, where appropriate, placed in the context of the portion of the lesson it affects. For each unit the information generally given are the following: 1) purpose, 2) entry points, 3) units joined, 4) units from which it can be entered, and 5) units to which it can exit. The BACK and TERM keys are not considered in the following as a way to enter or exit a unit.

Units:

STR1

welcome — loads character set and branches to the current unit returning from CSCOMMENT, otherwise jumps to the next unit.
entry point: ret
entry from: initial unit encountered
exits to: titre

titre — displays title page
joins: uarrow, rarrow, darrow, pressend
entry from: welcome
exits on selected key press to: comm, sansfin

comm — describes how to make comments
joins: comments, wait
entry from: titre
exits on selected key press to: exp, yy

choices — describes the three "modes"
entry from: comm, and any point in the lesson on "term" option
exits on selected key press to: term, choices

exp — describes how to use the NEXT, BACK, and TERM keys
joins: wait, pressend
entry from: choices
exits on selected key press to: term, choices

term	explains what the "term" options are joins: pressend entry from: exp, and any point in the lesson on "term" option exits to: the next unit or previous unit when entered via "term" option
more	introduces the lesson joins: wait, pressend entry from: term exits on selected key press to: dispa, term
dispa	shows the structure of the lesson entry from: more, and any point of the lesson on "term" option exits to: dispa2
dispa2	branch unit to the main parts of the lesson entry from: dispa exits on selected key press to: sansfin, intro, card, llist, stack, queue, deque, alloc, sequen, avail, linked
intro	general approach to lists joins: wait, pressend entry from: dispa2 exits on selected key press to: info, dispa
info	presents nodes and fields joins: concern, wait, fig, darrow entry from: intro exits on selected key press to: card, intro
card	represents a deck of cards as a list joins: wait, fig, pressend entry from: info exits on selected key press to: deck, info
deck	continuation of card joins: fig, wait, pressend entry from: card exits on selected key press to: display, card
display	explains how to refer to a given field within a node joins: wait, display2, nota-1, nota-2, algo, adl entry from: deck exits on selected key press to: newcard, yy, deck

newcard	presents an algorithm for placing a new card on the top of a pile entry point: neca joins: algo, waitn, depile, wait, stepl, step 2, adl entry from: display
llist	presents text on linear lists joins: wait, pressend entry from: newcard exits on selected key press to: opper, card
opper	explains how to access a node entry point: opper1 joins: wait, link1, k entry from: llist exits on selected key press to: noper, op1, llist
op1	shows the accessed node with a circle joins: uarrow, pressend entry from: opper exits on selected key press to: opins, llist
oppins	explains how to insert a node in a list entry point: opins1 joins: link1, wait, k entry from: op1 exits on selected key press to: nopins, one, op2, opper
op2	shows the insertion of a node in a list joins: ins11, pressend entry from: oppins exits on selected key press to: opdel, opper
opdel	explains how to delete a node from a list entry point: o1 joins: link1, wait, k entry from: op2 exits on selected key press to: nopdel, del12
del12	shows the deletion of a node joins: del11, pressend entry from: opdel exits on selected key press to: opcomb, oppins
opcomb	presents the combination of two linear lists joins: link1, wait, rarrow, node, pressend entry from: del12 exits on selected key press to: opsplit, opdel

opsplit — explains how to split a linear list
entry point: opsplitl
joins: linkl, wait, k
entry from: opcomb
exits on selected key press to: nopsplit, op4

op4 — shows the splitting of a linear list
joins: wait, pressend
entry from: opsplit
exits on selected key press to: opleft, opcomb

opleft — explains some operations one might want to perform on a linear list
joins: wait, pressend
entry from: op4
exits on selected key press to: stack, opsplit

stack — presents text on stacks
joins: wait, pressend
entry from: opleft, dispa2
exits on selected key press to: destack, llist

destack — shows a representation of a stack
joins: figstack, statop, starrow
entry from: stack
exits on selected key press to: popush, stack

popush — prepares the insertion and deletion of a node in a stack
joins: figstack, rarrow, starrow, shown
entry from: destack
exits to: jl

jl — chooses insertion or deletion
entry from: popush, pop, push
exits on selected key press to: queue, pop, push

pop — pops a node
entry points: neg, ero, po
joins: erarrow, topdo, topar, nodef, shown

push — pushes a node
entry points: null, pu
joins: erarrow, yopdo. topar, nodef, dar, shown
entry from: jl
exits to: jl

queue — presents text on queues
joins: wait, pressend
entry from: jl, dispa2
exits on selected key press to: desque, stack

desque
shows a graphic representation of a queue
joins: fboxes, darrow, pressend
entry from: queue
exits on selected key press to: ppqueue, queue

ppqueue
prepares the insertion or the deletion of a node in a queue
joins: track, dotdot, front, rear, shown
entry from: desque
exits to: qul

qul
chooses insertion or deletion
joins: next
entry from: ppqueue, empty, full, del-q, ins-q
exits on selected key press to: qul, deque, ins-q, del-q

ins-q
inserts a new item in a queue or in a deque (on the right)
entry points: la0, la1, la2, la3
joins: clean, cadel, waitn, cadef, rear, frontl, rightf, rightm, front, leftf, larrow, rarrow, rectan, shown

del-q
deletes an item from a queue or a deque (on the left)
entry points: dg0, dg1, dem, dg2, dg3
joins: clean, cadel, waitn, cadef, front, frontl, rear, rightf, leftm, larrow, rarrow, rectan, num, shown, reset
entry from: qul, dul
exits on selected key press to: empty, qul, dul

deque
presents text on deques
joins: wait, pressend
entry from: qul, dispa2
exits on selected key press to: desdeque, queue

desdeque
shows a graphic representation of a deque
joins: figdeque, pressend
entry from: deque
exits on selected key press to: ppdeque, deque

ppdeque
prepares the insertion or deletion of a node of a deque
joins: track, dotdot, leftm, rightm
entry from: desdeque
exits to: dul

dul — branches to next unit to test for emptiness or fullness
joins: next, show
entry from: dul, ppdeque
exits on selected key press to: alloc, yy, dul, side

side — chooses on which side deletion or insertion is to be performed
entry point: es
joins: next, insertt, deletee
entry from: dul
exits on selected key press to: full, empty, insleft, ins-q, del-q, delright, es

insleft — inserts an item on the left side of a deque
entry point: il0
joins: clean, cadel, waitn, cadef, leftm, leftf, lrarrow, rarrow, rectan, num, rightf, shown
entry from: side
exits on selected key press to: full, dul

delright — deletes an item on the right side of a deque
entry point: dr0
joins: clean, cadel, waitn, cadef, leftf, rightm, larrow, rarrow, rectan, num, shown, reset
entry from: side
exits on selected key press to: empty, dul

alloc — introduces storage allocation
joins: al, link4, link5
entry from: dispa2, dul
exits on selected key press to: insist, sequen, linked

sequen — introduces sequential allocation
joins: as, pressend
entry from: alloc
exits on selected key press to: seq2, alloc

seq2 — explains how to delete and insert a node in a stack and a queue
joins: as, pressend
entry from: sequen
exits on selected key press to: seq21, sequen

seq21 — additional information on sequential allocation
joins: wait, pressend
entry from: seq2
exits on selected key press to: avail, yy, seq2

avail — explains the use of the list of available space
joins: ava, pressend
entry from: seq21
exits on selected key press to: avail2, link3

avail2 — shows how to use the list of available space when inserting and deleting a node from a stack
joins: avb, figstack, rarrow, darrow, step, pressend
entry from: avail
exits on selected key press to: linked, sequen

linked — introduces linked allocation
joins: lp, link4, link5, display2, pressend
entry from: dispa2, alloc
exits on selected key press to: link3, avail

link3 — prepares to insert and to delete a node in a queue
entry points: j2, j21, j22
joins: link1, depress, delins, next
entry from: liked
exits on selected key press to: miss, insert, delete

insert — chooses after which node a new node is inserted
joins: delins, next
entry point: link3
exits to: j3

j3 — branches to ins1 or ins2
entry from: insert
exits on selected key press to: ins1, ins2

ins1 — inserts a new node
joins: nearer, ins11
entry from: j3

delete — chooses the node to be deleted
entry point: j4
joins: delins, next
entry from: link3
exits on selected key press to: del1, del2

dell — deletes the selected node
joins: nearer, delll
entry from: delete
exits to: j21

yy — implementation of jumpout to the lesson STR2
entry point: insiste
joins: schaus
entry from: termd, seq21, dul, newcard, display, choices
exits to: lesson STR2 or insiste if not enough memory available

sansfin — exits the lesson
entry from: any place in the lesson via "term" option
exits to: "choose a lesson" page

STR2

start2 — loads character set and branches to the current unit returning from CSCOMMENT, otherwise jumps to next unit
entry point: retr
entry from: initial unit encountered
exits to: idrill

ques — loads in n38-n41 code of the answer
entry from: d21, d41, d42, d43, d44, d45, d51, d52, d53, d54, d55, d56, d57, inter, drilld, drille
exits to: expp

expp — judge unit for the previous questions
entry points: exp2, rel, stuck, rb
joins: depress, sho, locate, errors, eral
entry from: ques
exits on selected key press to: exit-g, jul, idrill, erase, erasel, h-d2

exit-g — branches to the end of the drill
entry from: expp
exits on selected key press to: d23, d45, d57

jul — judges the answer of the student
entry point: ju2
joins: locate, press, arrow
entry from: expp
exits on selected key press to: h-d2, erase, erasel, return, ju2

erase
erases the last character typed
joins: eral
entry from: expp, jul
exits to: expp (rel)

erasel
erases all the answers of the student
joins: depress
entry from: expp, jul
exits to: expp (exp2)

h-d2
help sequence
entry point: 11
joins: errors, jd21, jd22, h-d211a
entry from: expp

idrill
list of the drills
entry from: expp and any place in the lesson via "term" option
exits on selected key press to: sansfint, drilla, drillb, drillc, drilld, drille

drilla
first drill
entry from: idrill
exits to: sep

sep
sets a permutation
entry from: drilla, fuite
exits to: fill

fill
displays three nodes filled with random numbers
joins: random, press
entry from: sep
exits on selected key press to: branch, drilla, exitl

branch
decides to stop to ask questions
joins: depress
entry from: ql, fill
exits on selected key press to: fuite, ql

fuite
ends the drill and displays the number of errors
joins: nextlab, rerrors
entry from: branch
exits on selected key press to: xx, xb, idrill, drillb, sep

ql
presents the questions to the students
entry points: a, c, qex
joins: press, errors, help2, random
entry from: branch
exits on selected key press to: helpl, exitl, branch

drillb — pressents the second drill
joins: skip, wait
entry from: idrill, fuite, inter, d2, 3, salir

d21 — asks the first question of the second drill
joins: jd21
entry from: drillb
exits to: quesll

inter — asks the second question of the second drill
joins: wait, larrow, jd22
entry from: return
exits to: quesl2

d23 — shows step2
joins: wait, rarrow, nextlab
entry point: exit2
entry from: return
exits on selected key press to: drillb, drillc, drilla, idrill, xx, xb

drillc — presents the third drill
entry points: retry 3, ici
joins: head, wait, skip, depressl
entry from: idrill, d23
exits on selected key press to: salir, a00

a00 — displays the questions and judges them
joins: show, text, al, a2, a3, a4, a5, press, help3, errors, newques
entry point: aa
entry from: drillc
exits on selected key press to: ici (drillc), exit3 (salir)

salir — displays the number of errors made on the third drill
entry point: exit 3
joins: wait, nextlab
entry from: drillc
exits on selected key press to: xx, xb, drillb, drilld, idrill

drilld — presents the fourth drill
joins: skip, assum, use, wait
entry from: idrill, salir, drillc, d45
exits to: ques2lla

d41, d42, d43, d44 — second, third, fourth, and fifth question of the fourth drill
entry from: return
exits to: ques2llb, ques2llc, ques2l2a, ques2l2b

d45 sixth question of the fourth drill
entry points: exit4, ret4
joins: depress, rerrors, nextlab
entry from: return, drilld
exits on selected key press to: xx, drilld, drille, idrill

drille presents the fifth drill
joins: skip, assum, use, wait
entry from: d45, d57, idrill
exits on selected key press to: xx, exit5, ques221c, drilld

d51, d52, d53, d54, d55, d56 second, third, fourth, fifth, sixth, seventh question of the fifth drill
entry from: return
exits on selected key press to: ques221b, ques221c, ques221d, ques222a, ques222b, ques222c

d57 eighth question of the fifth drill and displays the number of errors
entry points: ret5, exit5
joins: depress, rerrors, nextlab
entry from: return, drille
exits on selected key press to: xx, idrill, drilld, drille

xx implementation of jumpout to the lesson STR1
joins: schau
entry from: fuite, drillb, d23, salir, drilld, d45, drillc, d57, termterm, termm, termi
exits to: STR1 or xx if not enough memory available

xb implementation of jumpout to the lesson STR1
joins: schau
entry from: fuite, d23, salir
exits to: STR1 or xb if not enough memory available

sansfint exits the lesson
entry from: any place in the lesson via "term" option
exits to: "choose a lesson" page

```
 1  UNIT     WELCOME
 2  DEFINE   STUDENT
 3           IN2=N100
 4           TEST=N26
 5           MYNAME=N145
 6           OL=N24
 7           OR=N3
 8  TERM     C
 9  WRITE    ¬0AT,1022¬1(↑PLEASE STAND BY)                                    RET

10  ENTRY    RET
11  CHARSET  STR1,CHARSP
12  GOTO     ERROR=-1,RET,X
13  JOIN     MYNAME→↑7STR1↑7,COMEIN
14  JUMP     TITRE
15  *                                                                         TITRE

16  UNIT     TITRE
17  ERASE
18  SIZE     6
19  WRITE    ¬0AT,1207¬1↑S↑T↑R↑U↑C↑T↑U↑R↑E
20  WRITE    ¬0AT,47,319¬1↑S↑T↑R↑U↑C↑T↑U↑R↑E
21  SIZE     0
22  WRITE    ¬0AT,2423¬1 BY ↑MICHEL ↑J.↑S. ↑CHESNEL
23  SIZE     6
24  WRITE    ¬0AT,45,318¬1↑S↑T↑R↑U↑C↑T↑U↑R↑E
25  WRITE    ¬0AT,45,317¬1↑S↑T↑R↑U↑C↑T↑U↑R↑E
26  WRITE    ¬0AT,44,316¬1↑S↑T↑R↑U↑C↑T↑U↑R↑E
27  CATCHUP
28  SIZE     0
29  DRAW     805!862!1362!1305!805!SKIP!824!1324!SKIP!842!1342
30  CATCHUP
31  DELAY    .1
32  CIRCLE   75,183,304,180,360
33  CALC     OR→1333
34  JOIN     DARROW
35  CIRCLE   76,334,386,180,0
36  CALC     OR→852
37  JOIN     DARROW
38  CIRCLE   15,489,322,90,0
39  DRAW     1264!2364
40  CIRCLE   15,490,191,0,-90
41  DRAW     2162!2105
42  CIRCLE   15,31,191,-92,-180
43  DRAW     2003!1203
44  CIRCLE   13,29,323,90,185
45  CALC     OR→1105
46  JOIN     RARROW
47  CATCHUP
48  JOIN     PRESSEND
49  BACK     SANSFIN
50  NEXT     COMM
51  *
```

```
                                                                                                  COMM
UNIT    COMM
JOIN    COMMENTS
JOIN    WAIT
JUMP    KEY=NEXT,CHOICES,X
JUMP    KEY=BACK,TITRE,X
*
                                                                                                  CHOICES
UNIT    CHOICES
BACK    COMM
TERM    M
BASE
ERASE
WRITE   ¬0AT,1008¬1↑HOW DO YOU WANT TO GO THROUGH THIS LESSON↑/

             1- ↑TEXT ONLY, WITHOUT DRILLS.
             2- ↑DRILLS ONLY, WITHOUT TEXT.
             3- ↑NORMALLY, WITH BOTH TEXT AND DRILLS.
WRITE   ¬0AT,124,112¬1↑YOU CAN RETURN TO THIS FRAME
        ANY TIME BY PRESSING THE ↑T↑E↑R↑M
        KEY, TYPING ↑,M↑, (MODE) AND
        PRESSING THE ↑N↑E↑X↑T KEY.
DRAW    2416;2446;2846;2816;2416
ARROW   2030
LONG    1
MATCH   N26,1,2,3
JUDGE   N26,IGNORE,OK
CALC    N26=N26-1
PACK    N100,DRILLA
JUMP    N26,EXP,YY,EXP                  $$ DRILLA(STR2)
*
-------------------------------------------------------------------------------------------------------------------------------- INTRO
                                                                                                  SPACE LEFT = 17
                                                                                                  EXP
UNIT    EXP
ERASE
INHIBIT ERASE
WRITE   ¬0AT,811¬1↑USUALLY BY PRESSING THE ↑N↑E↑X↑T KEY YOU WILL
        GO TO THE NEXT PART OF THE FRAME.↑WHEN THE
        FRAME IS COMPLETED THE ↑N↑E↑X↑T KEY WILL LEAD
        YOU TO THE ↑N↑E↑X↑T FRAME.↑THE COMPLETION OF A
        FRAME WILL BE INDICATED BY THE POSITION OF
        THE -↑N↑E↑X↑T- MESSAGE AT THE RIGHT BOTTOM OF
        THE SCREEN.
JOIN    WAIT
GOTO    KEY=BACK,EXP,X
WRITE   ¬0AT,1611¬1↑THE ↑B↑A↑C↑K KEY WILL GET YOU TO THE BEGINNING
        OF THE CURRENT FRAME.↑WHEN AT THE END OF THE
        FRAME THE ↑B↑A↑C↑K KEY WILL GET YOU TO THE PRE-
        VIOUS FRAME.
JOIN    WAIT
GOTO    KEY=BACK,EXP,X
WRITE   ¬0AT,2111¬1↑WHEN YOU ARE LOST PUSH THE ↑T↑E↑R↑M KEY, THEN
        TYPE ↑,TERM↑, TO SEE THE ↑T↑E↑R↑M OPTIONS AVAIL-
        ABLE IN THIS LESSON OR HOW TO EXIT.
JOIN    PRESSEND
```

```
NEXT     TERM
BACK     CHOICES
*                                                                          TERM

UNIT     TERM
ERASE
NEXT     MORE
BACK     EXP
TERM     TERM
TERM     TERMS
WRITE    ¬0AT,809¬1+THE AVAILABLE +T+E+R+M OPTIONS ARE THE FOLLOWING+!
WRITE    ¬0AT,1315¬1-+,TERM+, TO GET TO THIS FRAME,

         -+,COMMENT+, TO MAKE A COMMENT ABOUT THE LESSON,

         -+,M+, TO CHOSE HOW TO GO THROUGH THE LESSON,

         -+,I+, TO SEE THE STRUCTURE OF THE LESSON,

         -+,D+, TO SEE THE LIST OF DRILLS,

         -+,O+, TO EXIT THE LESSON.
JOIN     PRESSEND
END
*                                                                          MORE

UNIT     MORE
NEXT     DISPA
BACK     TERM
INHIBIT ERASE
ERASE
WRITE    ¬0AT,1002¬1   +IN ORDER TO USE A COMPUTER PROPERLY, IT IS IMPORTANT TO
         ACQUIRE A GOOD UNDERSTANDING OF THE STRUCTURAL RELATIONSHIPS
         PRESENT WITHIN THE DATA, AND OF THE TECHNIQUES FOR REPRESENTING
         AND MANILPULATING SUCH STRUCTURE WITHIN A COMPUTER.

            +THE FOLLOWING LESSON SUMARIZES THE MOST IMPORTANT FACTS
         ABOUT INFORMATIONS STRUCTURES+!
JOIN     WAIT
GOTO     KEY=BACK,MORE,X
WRITE    ¬0AT,1907¬1- THE STATIC AND DYNAMIC PROPERTIES OF DIFFERENT KINDS OF
           STRUCTURES

         - MEANS FOR STORAGE ALLOCATION AND REPRESENTATION OF
           STRUCTURES,

         - EFFICIENT ALGORITHMS FOR CREATING, ALTERATING, ACCESSING
           AND DESTROYING STRUCTURAL INFORMATION.
JOIN     PRESSEND
*                                                                          DISPA

UNIT     DISPA
ERASE
TERM     I
BASE
INHIBIT ERASE
WRITE    ¬0AT,716¬11- +INTRODUCTION
```

```
157          2- ↑EXAMPLE
158          3- ↑LINEAR LISTS
159                  A. STACKS
160                  B. QUEUES
161                  C. DEQUES
162                  D. STORAGE ALLOCATION
163                       E. SEQUENTIAL ALLOCATION
164                       F. LIST OF AVAILABLE SPACE
165                       G. LINKED ALLOCATION
```

CONTENTS

SPACE LEFT = 20

```
166  WRITE   ¬0AT,1815¬1↑PRESS 1,2,3,A,B,C,D,E,F OR G TO SEE
167               ANY OF THESE SECTIONS.
168  WRITE   ¬0AT,2808¬1↑YOU CAN RETURN TO THIS INDEX AT ANY TIME BY PRESSING
169          THE ↑T↑E↑R↑M KEY, TYPING I(I AS INDEX), AND PRESSING ↑N↑E↑X↑T
170  WRITE   ¬0AT,417¬1↑T↑Y↑P↑E 0 ↑I↑F ↑Y↑O↑U ↑W↑A↑N↑T ↑T↑O ↑Q↑U↑I↑T
171  DRAW    316!343!443!416!316
172  JUMP    DISPA2
173  *
```

DISPA2

```
174  UNIT    DISPA2
175  BASE    DISPA
176  BACK    MORE
177  INHIBIT ERASE
178  AT      2330
179  ERASE   3
180  ARROW   2330
181  LONG    1
182  MATCH   V1,0,1,2,3,A,B,C,D,E,F,G,H,Z
183  JUDGE   V1,IGNORE,OK
184  JUMP    V1,X,SANSFIN,INTRO,CARD,LLIST,STACK,QUEUE,DEQUE,ALLOC,SEQUEN,AVAIL,LINKED,X,X
185  *
```

INTRO

```
186  UNIT    INTRO
187  BACK    DISPA
188  NEXT    INFO
189  BASE
190  ERASE
191  WRITE   ¬0AT,302¬1   ↑COMPUTER PROGRAMS USUALLY OPERATE ON TABLES OF INFORMATION.
192          ↑IN MOST CASES THESES TABLES ARE NOT SIMPLY AMORPHOUS MASSES OF
193          NUMERICAL VALUES, THEY INVOLVE IMPORTANT STRUCTURAL RELATION=
194          =SHIPS BETWEEN THE DATA ELEMENTS.
195             ↑IN ITS SIMPLEST FORM, A TABLE MIGHT BE A LINEAR LIST OF
196          ELEMENTS AND ONE CAN ASK SUCH QUESTIONS AS↑!
197  JOIN    WAIT
198  GOTO    KEY=BACK,INTRO,X
199  AT      1108
200  WRITE   ¬0AT,1108¬1- WHICH ELEMENT IS FIRST IN THE LIST,
201
202          - WHICH ELEMENT IS LAST IN THE LIST,
203
204          - WHICH ELEMENTS PRECEDE AND FOLLOW A GIVEN ONE
205  JOIN    WAIT
206  GOTO    KEY=BACK,INTRO,X
207  WRITE   ¬0AT,1802¬1   ↑IN MORE COMPLICATED SITUATIONS, THE TABLE MIGHT BE↑!
208  JOIN    WAIT
```

```
GOTO    KEY=BACK,INTRO,X
WRITE   -0AT,2008-1- A TWO DIMENSIONAL ARRAY.

      / - A N-DIMENSIONAL ARRAY FOR HIGHER VALUES OF N.

        - A TREE STRUCTURE REPRESENTING HIERARCHICAL OR
          BRANCHING RELATIONSHIP.

        - A COMPLEX MULTILINKED STRUCTURE WITH A GREAT MANY
          INTERCONNECTIONS SUCH AS WE MAY FIND IN A HUMAN BRAIN.
JOIN    PRESSEND
*                                                                    CONCERN

UNIT    CONCERN
AT      703
WRITE      +OUR CONCERN WILL BE ALMOST ENTIRELY WITH THE STRUCTURE AS
        REPRESENTED INSIDE A COMPUTER! THE CONVERSION FROM EXTERNAL
        TO INTERNAL REPRESENTATION WILL NOT BE DISCUSSED.
*                                                                    INFO

UNIT    INFO
NEXT    CARD
BACK    INTRO
ERASE
JOIN    CONCERN
JOIN    WAIT
GOTO    KEY=BACK,INFO,X
AT      1503
WRITE      +≠≤+≡ +THE INFORMATION IN A TABLE CONSISTS OF A SET OF NODES
             (ALSO CALLED +,TERMS, ELEMENTS, RECORDS, ENTITIES ...+,).
JOIN    WAIT
GOTO    KEY=BACK,INFO,X
MODE    ERASE
JOIN    CONCERN
MODE    WRITE
DELAY   .5
AT      1803
WRITE      +≠≤+≠≤+≡ +EACH NODE CONSISTS OF ONE OR MORE CONSECUTIVE WORDS OF
             THE COMPUTER MEMORY.
JOIN    WAIT
GOTO    KEY=BACK,INFO,X
FIGURE  712,752,1052,1012,712

-------------------------------------------------------------------- DECK
                                                          SPACE LEFT =: 11

JOIN    WAIT
GOTO    KEY=BACK,INFO,X
AT      2103
WRITE      +≠≤+≡ +EACH NODE IS DIVIDED IN FIELDS.
JOIN    WAIT
GOTO    KEY=BACK,INFO,X
CALC    OR+712
JOIN    FIG
AT      1213
WRITE   FIELD 1    FIELD 2    FIELD 3    .......
JOIN    WAIT
GOTO    KEY=BACK,INFO,X
```

```
AT        2303
WRITE        ↑≠S↑≡ ↑THE ADDRESS OF A NODE ( ALSO CALLED THE LINK, REFERENCE
               OR POINTER) IS THE MEMORY LOCATION OF ITS FIRST WORD.
JOIN      WAIT
GOTO      KEY=BACK,INFO,X
AT        414
WRITE     ↑P↑O↑I↑N↑T↑E↑R
LINE      417,717
CALC      UK←717
JOIN      DARROW
JOIN      WAIT
GOTO      KEY=BACK,INFO,X
AT        2603
WRITE        ↑≠S↑≡ ↑THE CONTENTS OF ANY FIELD WITHIN A NODE MAY REPRESENT
               NUMBERS, ALPHABETIC CHARACTERS, LINKS OR ANYTHING ELSE.
AT        3220
WRITE     ¬0AT,3233¬1 ↑PRESS -↑N↑E↑X↑T- TO SEE AN EXAMPLE.
*
                                                                                CARD
UNIT      CARD
BACK      INFO
NEXT      DECK
BASE
ERASE
WRITE     ¬0AT,616¬1↑LET US SUPPOSE WE WISH TO REPRESENT
WRITE     ¬0AT,724¬1A PILE OF CARDS↑!
JOIN      WAIT
GOTO      KEY=BACK,CARD,X
WRITE     ¬0AT,1010¬1↑- EACH CARD WILL BE REPRESENTED BY A NODE.

          ↑- EACH NODE MIGHT HAVE FOUR FIELDS.
DELAY     .7
WRITE     ¬0AT,1713¬1↑T↑A↑G


          ↑S↑U↑I↑T


          ↑R↑A↑N↑K


          ↑N↑E↑X↑T
DELAY     .5
JOIN      WAIT
GOTO      KEY=BACK,CARD,X
WRITE     ¬0AT,1725¬1TAG=1 MEANS THE CARD IS FACE DOWN,
          TAG=0 MEANS THE CARD IS FACE UP.

          SUIT=1,2,3 OR 4 FOR CLUBS,DIAMONDS
          HEARTS OR SPADES.

          RANK=1,2,   .... ,13 FOR ACE,TWO...
          KING.

          NEXT IS A LINK TO THE CARD BELOW.
DELAY     .4
```

```
MODE     REWRITE
WRITE    ¬0AT,610¬1+THE REPRESENTATION OF A NODE WILL BE THE FOLLOWING
AT       724
ERASE    16
AT       1010
ERASE    1+5
CALC     OR+1015
JOIN     FIG
DELAY    .7
MODE     WRITE
AT       1218
WRITE    ¬0AT,1218¬1+T+A+G        +S+U+I+T        +R+A+N+K        +N+E+X+T
JOIN     PRESSEND
*
                                                                        DECK
UNIT     DECK
PACK     N100,DRILLA
NEXT     DISPLAY
BACK     CARD
ERASE
SIZE     1
WRITE    ¬0AT,708¬1+F+O+R +T+H+R+E+E +C+A+R+D+S+1
SIZE     0
CALC     OR+809
JOIN     FIG
DELAY    .3
CALC     OR+1611
JOIN     FIG
DELAY    .4
CALC     OR+2413
JOIN     FIG
JOIN     WAIT
GOTO     KEY=BACK,DECK,X
WRITE    ¬0AT,748¬1ADDRESS (MEMORY)
             101
WRITE    ¬0AT,1654¬1786
WRITE    ¬0AT,2456¬1342
WRITE    ¬0AT,1202¬1CARD+1  FACE  DOWN   CLUBS      KING
WRITE    ¬0AT,2004¬1CARD+1  FACE   UP   HEARTS      TWO
WRITE    ¬0AT,2806¬1CARD+1  FACE   UP  DIAMONDS     ACE
DELAY    .4
--------------------------------------------------------------------- DISPLAY
                                                            SPACE LEFT = 47
WRITE    ¬0AT,3225¬1 +PRESS -+N+E+X+T- TO FILL THE  FIELDS.
JOIN     WAIT
GOTO     KEY=BACK,DECK,X
AT       3220
ERASE    35
WRITE    ¬0AT,1014¬11          1          13        ≥JI
WRITE    ¬0AT,1816¬10          4          2        101
WRITE    ¬0AT,2618¬10          2          1        786
WRITE    ¬0AT,112¬1+ +THE +GREEK LAMBDA IS USED TO DENOTE THE NULL LINK
           (THE LINK TO NO NODE), THE LINK ≥JI≥ APPEARS IN NODE
           101 SINCE THE KING OF CLUBS IS THE BOTTOM CARD OF
           THE PILE.
```

```
370  JOIN     PRESSEND
371  *
                                                                          DISPLAY2
372  UNIT     DISPLAY2
373  AT       1405
374  WRITE    ↑T↑O↑P
375  LINE     1311,1315
376  CALC     OR↑1315
377  JOIN     RARROW
378  CALC     OR↑1215
379  JOIN     NODE
380  CIRCLE   2↑166↑304
381  LINE     1322,1327
382  CALC     OR↑1327
383  JOIN     RARROW
384  CALC     OR↑1227
385  JOIN     NODE
386  CIRCLE   2↑262↑304
387  LINE     1334,1339
388  CALC     OR↑1339
389  JOIN     RARROW
390  CALC     OR↑1239
391  JOIN     NODE
392  CIRCLE   2↑358↑304
393  FIGURE   1346,1351↑1451
394  WRITE    ¬0AT,1550¬12KL
395  *
                                                                          NOTA-1
396  UNIT     NOTA-1
397  AT       1803
398  WRITE       ↑N↑S↑6O↑S↑6T↑S↑6A↑S↑6T↑S↑6I↑S↑6O↑S↑6N↑6S↑S↑6↑!
399
400            ↑REFERING TO FIELDS WITHIN NODES IS DONE SIMPLY BY GIVING
401          THE NAME OF THE FIELD FOLLOWED BY A LINK TO THE DESIRED
402          NODE IN PARENTHESIS↑!
403  *
                                                                          NOTA-2
404  UNIT     NOTA-2
405  AT       2403
406  WRITE          ↑S↑U↑I↑T(↑T↑O↑P)= 2  (DIAMONDS)
407                 ↑R↑A↑N↑K(101)=13  (KING)
408
409          OR THROUGH LINK FIELDS IN OTHER NODES↑!
410
411                 ↑T↑A↑G(↑N↑E↑X↑T(↑T↑O↑P))= 0   (FACE UP)
412  *
                                                                          ALGO
413  UNIT     ALGO
414  WRITE    ¬0AT,2312¬1   ↑PRESS ¬↑N↑E↑X↑T¬ TO SEE A SIMPLE ALGORITHM
415          FOR PLACING A NEW CARD ON TOP OF THE PILE
416          ASSUMING ↑N↑E↑W↑C↑A↑R↑D IS A LINK VARIABLE WHOSE
417          VALUE IS A LINK TO THE NEW CARD
418  DRAW     2111↑2154↑2754↑2711↑2111
419  *
                                                                          DISPLAY
420  UNIT     DISPLAY
421  BACK     DECK
```

```
NEXT     TEST,NEWCARD,YY,YY                  $$  DRILLA(STR2)
ERASE
AT       503
WRITE       ↑WHEN DISPLAYING COMPUTER REPRESENTATION OF NODES IT IS
         USUALLY CONVENIENT TO REPRESENT LINKS BY ARROWS, SO OUR
         EXAMPLE WOULD APPEAR↑!
JOIN     WAIT
GOTO     KEY=BACK,DISPLAY,X
JOIN     DISPLAY2
DELAY    .7
JOIN     WAIT
GOTO     KEY=BACK,DISPLAY,X
JOIN     NOTA-1
JOIN     WAIT
GOTO     KEY=BACK,DISPLAY,X
JOIN     NOTA-2
DELAY    .5
JOIN     PRESS
JOIN     WAIT
GOTO     KEY=BACK,DISPLAY,X
MODE     ERASE
JOIN     NOTA-1
JOIN     NOTA-2
MODE     WRITE
JOIN     TEST,ALGO,AD1
*
-------------------------------------------------------------------------------------------------------------------------------- NEWCARD
                                                                                                                  SPACE LEFT = 59
                                                                                                                        STEP1
UNIT     STEP1
WRITE    ¬0AT,3225¬1 ↑PRESS -↑N↑E↑X↑T- TO DO STEP ONE.
JOIN     WAIT
GOTO     KEY=BACK,NECA,X
AT       1128
ERASE    10
AT       1228
ERASE    2
LINE     1028,1228
CIRCLE   2,207,416
LINE     627,629
CIRCLE   40,224,376,90,-90
CALC     OR→1128
JOIN     LARROW
AT       936
WRITE    ↑T↑O↑P
*
                                                                                                                        STEP2
UNIT     STEP2
WRITE    ¬0AT,3225¬1 ↑PRESS -↑N↑E↑X↑T- TO DO STEP TWO.
JOIN     WAIT
GOTO     KEY=BACK,NECA,X
AT       934
ERASE    6
AT       219
ERASE    160
LINE     520,528
```

```
474  AT       610
475  WRITE    ↑T↑O↑P
476  LINE     614,620
477  CALC     OR↑620
478  JOIN     RARROW
479  JOIN     PRESSEND
480  *                                                                    NEWCARD

481  UNIT     NEWCARD
482  PACK     N100,DRILLR
483  NEXT     TEST,LLIST,YY,YY                SS DRILLA(STR2)
484  BACK     DISPLAY
485  ERASE
486  JOIN     TEST,X,ALGO
487  JOIN     TEST,X,WAITN                                              NECA

488  ENTRY    NECA
489  ERASE
490  AT       3002
491  WRITE    ↑SIMPLE ALGORITHM FOR PLACING A NEW CARD ON THE TOP OF THE PILE
492  JOIN     DEPILE
493  JOIN     WAIT
494  GOTO     KEY=BACK,NECA,X
495  WRITE    ¬0AT,1532¬1↑STEP 1 - ↑SET ↑N↑E↑X↑T(↑N↑E↑W↑C↑A↑R↑D) @ ↑T↑O↑P
496
497                 THIS SETS THE APPROPRIATE
498                 LINK IN THE NEW CARD.
499  DELAY    .5
500  JOIN     STEP1
501  JOIN     WAIT
502  GOTO     KEY=BACK,NECA,X
503  WRITE    ¬0AT,2232¬1↑STEP 2 - ↑SET ↑T↑O↑P @ ↑N↑E↑W↑C↑A↑R↑D
504
505                 THIS KEEPS ↑T↑O↑P POINTING
506                 TO THE TOP OF THE PILE.
507  JOIN     STEP2
508  JOIN     TEST,X,WAITN
509  JOIN     TEST,X,AD1
510  *                                                                    AD1

511  UNIT     AD1
512  ERASE
513  AT       2020
514  WRITE    ↑PRESS -↑N↑E↑X↑T- TO ANSWER
515           FEW QUESTIONS ABOUT THE
516           PREVIOUS EXAMPLE.
517  DRAW     1819!1844!2344!2319!1819
518  *                                                                    LLIST

519  UNIT     LLIST
520  NEXT     OPPER
521  BACK     CARD
522  BASE
523  ERASE
524  SIZE     5
525  WRITE    ¬0AT,502¬1↑LINEAR LISTS↑!
526  DELAY    .3
```

```
WRITE    ¬0AT,12,432¬1↑LINEAR LISTS
SIZE
WRITE    ¬0AT,1203¬1IS A SET OF N>0, NODES ↑X(1), ↑X(2)...... ↑X(N) WHOSE STUCTURAL
         PROPERTIES ESSENTIALLY INVOLVE ONLY THE LINEAR (ONE-DIMEN-
         SIONAL) RELATIVE POSITIONS OF THE NODES.
JOIN     WAIT
GOTO     KEY=BACK,LLIST,X
WRITE    ¬0AT,1708¬1        ↑IF N>0↑!


         - ↑X(1) IS THE FIRST NODE,

         - WHEN 1<K<N, THE KTH NODE ↑X(K) IS PRECEDED BY ↑X(K-1)
           AND FOLLOWED BY ↑X(K+1),

         - ↑X(N) IS THE LAST NODE.
JOIN     PRESSEND
*
```

---------- OPERATIONS

SPACE LEFT = 40

K

```
UNIT     K
WRITE    ¬0AT,2324¬1↑CHOSE A VALUE FOR K
AT       2432
ERASE    3
ARROW    241,121
LONG     1
SPECS    NOOKNO
*
```

OPPER

```
UNIT     OPPER
BACK     LLIST
INHIBIT  ERASE
ERASE
WRITE    ¬0AT,505¬1↑THE OPERATIONS WE MIGHT WANT TO PERFORM ON LINEAR
                 LISTS INCLUDE THE FOLLOWING↑!
JOIN     WAIT
GOTO     KEY=BACK,OPPER,X
JOIN     LINK1
WRITE    ¬0AT,2806¬11- ↑GAIN ACCESS TO THE KTH NODE OF THE LIST TO EXAMINE
            AND/OR CHANGE THE CONTENTS OF THE FIELDS
JOIN     WAIT
GOTO     KEY=BACK,OPPER,X
```

OPPER1

```
ENTRY    OPPER1
INHIBIT  ERASE
JOIN     K
MATCH    V1,1,2,3,4,5
JUMP     V1,NOPER,OP1
*
```

OP1

```
UNIT     OP1
NEXT     OPINS
BACK     LLIST
ADD1     V1
CIRCLE   2,278,128,14,360
```

```
FIGURE  2436,2457,2157,2097+12*V1,1897+12*V1
CALC    OR*1897+12*V1
JOIN    UARROW
CIRCLE  48,96*V1-32,256
JOIN    PRESSEND
*
                                                                                OPINS
UNIT    OPINS
BACK    OPPER
INHIBIT ERASE
ERASE
WRITE   ¬0AT,2806¬12¬ +INSERT A NEW NODE JUST BEFORE THE KTH NODE.
JOIN    LINK1
JOIN    WAIT
GOTO    KEY=BACK,OPINS,X
                                                                            OPINS1
ENTRY   OPINS1
INHIBIT ERASE
JOIN    K
MATCH   V2,1,2,3,4,5
JUMP    V2,NOPINS,ONE,OP2
*
                                                                                OP2
UNIT    OP2
NEXT    OPDEL
BACK    OPPER
CALCC   V2=0,V36*40,V36*30+V2
CALCC   V2=5,V46*45,V46*31+V2
JOIN    INS11
JOIN    PRESSEND
*
                                                                               NOPER
UNIT    NOPER
INHIBIT ERASE
CALC    V5*-1
JOIN    NOGEN
JUMP    OPPER1
*
                                                                                ONE
UNIT    ONE
INHIBIT ERASE
WRITE   ¬0AT,2605¬1THERE IS NOT ENOUGH ROOM ON THE SCREEN. SORRY ABOUT THAT.
JOIN    PRESS
PAUSE
JOIN    DEPRESS
AT      165,121
ERASE   13
AT      2605
ERASE   57
JUMP    OPINS1
*
                                                                              NOGEN
UNIT    NOGEN
AT      165,121
WRITEC  V5,(1 TO 5),(1 TO 4),(2 TO 5)
AT      241,121
ERASE   3
```

```
                                                                                                                NOPINS
UNIT     NOPINS
INHIBIT  ERASE
CALC     V5→1
JOIN     NOGEN
JUMP     OPINS1
*
                                                                                                                OPDEL
UNIT     OPDEL
BACK     OPINS
INHIBIT  ERASE
ERASE
WRITE    ¬0AT,2806¬13- ↑DELETE THE KTH NODE.
JOIN     LINK1
JOIN     WAIT
GOTO     KEY=BACK,OPDEL,X
                                                                                                  01
ENTRY    01
INHIBIT  ERASE
JOIN     K
MATCH    V2,1,2,3,4,5
CALCC    V2=0,V36→40,V36→30+V2
CALCC    V2=4,V46→45,V46→32+V2
JUMP     V2,NOPDEL,DEL12
*
                                                                                                                DEL12
UNIT     DEL12
NEXT     OPCOMB
BACK     OPINS
JOIN     DEL11
JOIN     PRESSEND
*
                                                                                                                NOPDEL
UNIT     NOPDEL
INHIBIT  ERASE
CALC     V5→-1
JOIN     NOGEN
JUMP     01
*
-------------------------------------------------------------------------------------------------------------------------------- SPLIT
                                                                                                                SPACE LEFT = 50
                                                                                                                OPCOMB
UNIT     OPCOMB
NEXT     OPSPLIT
BACK     OPDEL
ERASE
AT       2806
WRITE    ¬0AT,2806¬14- ↑COMBINE TWO (OR MORE) LINEAR LIST INTO A SINGLE LIST.
JOIN     LINK1
JOIN     WAIT
GOTO     KEY=BACK,OPCOMB,X
WRITE    ¬0AT,1913¬1↑F↑I↑R↑S↑IV2
DRAW     1915;2115;2117
CALC     OR→2117
JOIN     RARROW
```

```
CALC     OR*2017
JOIN     NODE
WRITE    ¬0AT,130,169¬1ITEMV2V1
CIRCLE   2,183,176,13,360
DRAW     2124:2129
CALC     OR*2129
JOIN     RARROW
CALC     OR*2029
JOIN     NODE
WRITE    ¬0AT,226,169¬1ITEMV2V2
CIRCLE   2,278,176,13,360
DRAW     2136:2141
CALC     OR*2141
JOIN     RARROW
CALC     OR*2041
JOIN     NODE
WRITE    ¬0AT,321,169¬1ITEMV2V3
CIRCLE   2,375,176,13,360
DRAW     2148:2152:2352
WRITE    ¬0AT,2351¬12KL
JOIN     WAIT
GOTO     KEY=BACK,OPCOMB,X
AT       1862
ERASE    2
AT       1913
ERASE    6
AT       2015
ERASE    4
LINE     2017,2020
AT       2115
ERASE    1
DRAW     1663:1863:1813:2113:2117
JOIN     PRESSEND
*
                                                                    OPSPLIT
UNIT     OPSPLIT
BACK     OPCOMB
INHIBIT  ERASE
ERASE
WRITE    ¬0AT,2806¬15- *SPLIT A LINEAR LIST INTO TWO (OR MORE) LISTS AFTER
            KTH NODE.
JOIN     LINK1
JOIN     WAIT
GOTO     KEY=BACK,OPSPLIT,X
                                                                    OPSPLIT1
ENTRY    OPSPLIT1
INHIBIT  ERASE
JOIN     K
MATCH    V2,1,2,3,4,
JUMP     V2,NOPSPLIT,OP4
*
                                                                    NOPSPLIT
UNIT     NOPSPLIT
INHIBIT  ERASE
CALC     V5*0
JOIN     NOGEN
JUMP     OPSPLIT1
```

OP4

```
UNIT    OP4
NEAT    OPLEFT
BACK    OPCOMB
JOIN    WAIT
GOTO    KEY=BACK,OPSPLIT,X
AT      2322
ERASE   110
ADD1    V2
AT      1602+V2≠12
ERASE   1
AT      1300+V2≠12
WRITE   ↑F↑I↑R↑S↑Tv2
LINE    1602+V2≠12,1802+V2≠12
WRITE   ¬0AT,1801+V2≠12¬12KL
LINE    1603+V2≠12,1603+V2≠12
JOIN    PRESSEND
*
```

OPLEFT

```
UNIT    OPLEFT
NEAT    STACK
BACK    OPSPLIT
ERASE
AT      812
WRITE   ↑OTHERS OPERATIONS ON LINEAR LISTS MIGHT
                BE THE FOLLOWING↑!
JOIN    WAIT
GOTO    KEY=BACK,OPLEFT,X
AT      1506
WRITE   6- ↑MAKE A COPY OF A LINEAR LIST,

        7- ↑DETERMINE THE NUMBER OF NODES IN A LIST,

        8- ↑SORT THE NODES OF THE LIST INTO ASCENDING ORDER BASED
           ON CERTAIN FIELDS OF THE NODES,

        9- ↑SEARCH THE LIST FOR THE OCCURRENCE OF A NODE WITH A
           PARTICULAR VALUE IN SOME FIELD.
JOIN    PRESSEND
*
```

---------- STACK

SPACE LEFT = 58

STACK

```
UNIT    STACK
NEXT    DESTACK
BACK    LLIST
BASE
ERASE
SIZE    2
AT      202
WRITE   A STACK↑!
SIZE
AT      403
WRITE   IS A LINEAR LIST FOR WHICH ALL INSERTIONS AND DELETIONS (AND
        USUALLY ALL ACCESSES) ARE MADE AT ONE END OF THE LIST
```

```
781             IN THE LIST, I.E. THE ONE WHICH HAS BEEN INSERTED
782   DELAY     .4
783   JOIN      WAIT
784   GOTO      KEY=BACK,STACK,X
785   AT        1011
786   WRITE     * ↑THE ITEM REMOVED IS THE YOUNGEST ONE CURRENTLY
787                 IN THE LIST, I.E. THE ONE WHICH HAS BEEN
788                 INSERTED MORE RECENTLY THAN ANY OTHER.
789
790             * ↑THE BOTTOM OF THE STACK IS THE LEAST ACCESSIBLE
791                 ITEM AND IT WILL NOT BE REMOVED UNTIL ALL
792                 THE OTHER ITEMS HAVE BEEN DELETED.
793
794             * ↑PUSH DOWN AND POP UP ARE OFTEN USED INSTEAD OF
795                 INSERT AND DELETE, RESPECTIVELY.
796
797             * ↑IN FACT NOTHING IS PHYSICALLY PUSHED DOWN,
798                 ITEMS ARE ADDED ONTO THE TOP.
799
800             * ↑STACKS ARE ALSO CALLED PUSH DOWN LISTS OR
801                 LAST IN FIRST OUT↑: ↑L↑I↑F↑O.
802
803             * ↑STACKS ARE USEFUL FOR THE PROCESSING OF LANGUAGES
804                 WITH A NESTED STRUCTURE, IN GENERAL STACKS
805                 OCCUR IN CONNECTION WITH A RECURSIVE ALGORITHMS .
806   JOIN      PRESSEND
807   *
808   UNIT      DESTACK
809   NEXT      POPUSH
810   BACK      STACK
811   ERASE
812   SIZE      2
813   AT        3125
814   WRITE     ↑STACK
815   SIZE
816   DELAY     .3
817   JOIN      FIGSTACK
818   JOIN      STATOP
819   JOIN      STARROW
820   AT        737
821   WRITE     ↑INSERT OR DELETE
822   AT        603
823   WRITE     TOP OF THE STACK
824   AT        1130
825   WRITE     ↑NEXT TO TOP
826   AT        1630
827   WRITE     ↑THIRD FROM TOP
828   AT        2130
829   WRITE     ↑FOURTH FROM TOP
830   AT        2630
831   WRITE     ↑BOTTOM
832   DELAY     .3
833   AT        314
834   WRITE     ↑PRESS -↑N↑E↑X↑T- TO POP OR PUSH THE STACK
835   FIGURE    113,152,452,413,113
836   *
```

```
UNIT     POPUSH
TERM     S
INHIBIT  ERASE
ERASE
JOIN     FIGSTACK
WRITE    ¬0AT,612¬1↑T↑O↑P
DRAW     616:620
CALC     OR←620
JOIN     RARROW
JOIN     STARROW
DRAW     2724:2924
WRITE    ¬0AT,3023¬12KL
CALC     N2←5
WRITE    ¬0AT,231¬1↑PRESS ↑↓U↑↓ TO POP UP THE STACK
         ↑PRESS ↑↓D↑↓ TO PUSH DOWN THE STACK
WRITE    ¬0AT,1836¬1↑PRESS ¬↑N↑E↑X↑T¬ TO LEAVE
DRAW     1635:1658:1958:1935:1635
WRITE    ¬0AT,281,224¬1↑PRESS ¬↑N↑E↑X↑T¬ TO LEAVE
JOIN     SHOWN
JUMP     J1
*
```

POP.PUSH

SPACE LEFT =: 75

J1

```
UNIT     J1
BASE     POPUSH
BACK     DESTACK
INHIBIT  ERASE
AT       540
ERASE    3
ARROW    540
LONG     1
ANSWER
CATCHUP
JUMP     QUEUE
MATCH    V1,U,D
JUMP     V1,J1,POP,PUSH
*
```

TOPDO

```
UNIT     TOPDO
WRITE    ¬0AT,2712¬1↑T↑O↑P
DRAW     2716:2724
*
```

POP

```
UNIT     POP
INHIBIT  ERASE
JOIN     ERARROW
CALC     N2←N2-1
GOTO     N2,X,NEG
CALC     N2←0
WRITE    ¬0AT,731¬1↑SORRY, THE STACK IS EMPTY
         TRY TO PUSH DOWN.
JOIN     PO
```

NEG

```
ENTRY    NEG
```

```
CALC     N5←N2
MODE     ERASE
JOIN     TOPAR
JOIN     NODEF
GOTO     N2=0,ERO,X
JOIN     DAR
MODE     WRITE
DRAW     3023-500*N2!3024-500*N2

ENTRY    ERO
MODE     WRITE
CALC     N5←N2-1
JOIN     N2=0,X,TOPAR
JOIN     N2=0,TOPDO,X
AT       731
WRITEC   N2=0¬,↑THE STACK IS EMPTY.¬,¬,

ENTRY    PO
JOIN     SHOWN
JUMP     J1
*

UNIT     PUSH
INHIBIT  ERASE
JOIN     ERARROW
GOTO     N2=5,NULL,X
CALC     N2←N2+1
MODE     ERASE
JOIN     N2=1,TOPDO,X
CALC     N5←N2-2
JOIN     N2=1,X,TOPAR
MODE     WRITE
CALC     N5←N2-1
JOIN     TOPAR
JOIN     NODEF
JOIN     N2=1,X,DAR
GOTO     PU

ENTRY    NULL
WRITE    ¬0AT,731¬1↑SORRY, THE STACK IS FULL.
         ↑POP UP IF YOU WANT↑.

ENTRY    PU
JOIN     SHOWN
JUMP     J1
*

UNIT     TOPAR
WRITE    ¬0AT,2612-500*N5¬1↑T↑O↑P
DRAW     2616-500*N5!2620-500*N5
CALC     OR←2620-500*N5
JOIN     RARROW
*

UNIT     DAR
DRAW     2724-500*N5!3024-500*N5
CALC     OR←3024-500*N5
```

ERO

PO

PUSH

NULL

PU

TOPAR

DAR

```
       *
                                                              NODEF
       UNIT     NODEF
       CALC     OR+252U-500*N5
       JOIN     NODE
       *
                                                              ERARROW
       UNIT     ERARROW
       AT       731
       ERASE    53
       AT       549
       ERASE    3
       *
                                                              SHOWN
       UNIT     SHOWN
       MODE     REWRITE
       WRITE    ¬0AT,2U2¬1+NUMBER OF NODES = ¬0S,N2¬1
       MODE     WRITE
       *
                                                              STARROW
       UNIT     STARROW
       LINE     724,1024
       LINE     1224,1524
       LINE     1724,2024
       LINE     2224,2524
       CALC     OR+1024
       JOIN     DARROW
       CALC     OR+1524
       JOIN     DARROW
       CALC     OR+2024
       JOIN     DARROW
       CALC     OR+2524
       JOIN     DARROW
       *
                                                              STATOP
       UNIT     STATOP
       LINE     635,628
       CALC     OR+628
       JOIN     LARROW
       *
--------------------------------------------------------------------- QUEUE
                                                       SPACE LEFT = 4
                                                              QUEUE
       UNIT     QUEUE
       NEXT     DESQUE
       BACK     STACK
       BASE
       ERASE
       SIZE     2
       AT       702
       WRITE    A QUEUE+!
       SIZE
       AT       903
       WRITE    IS A LINEAR LIST FOR WHICH ALL INSERTIONS ARE MADE AT ONE END
                OF THE LIST! ALL DELETIONS (AND USUALLY ALL ACCESSES) ARE MADE
```

```
          AT THE OTHER END
DELAY     .5
JOIN      WAIT
GOTO      KEY=BACK,QUEUE,X
AT        1611
WRITE     * ↑THE OLDEST ITEM IS ALWAYS REMOVED FIRST,
               THE NODES LEAVE THE LIST IN THE SAME
               ORDER AS THEY ENTERED IT.

          * ↑THE NODES ENTER AT THE REAR AND ARE
              REMOVED WHEN THEY ULTIMATELY REACH
              THE FRONT POSITION.

          * ↑QUEUES ARE ALSO CALLED FIRST IN FIRST
              OUT↑! ↑F↑I↑F↑O.
JOIN      PRESSEND
*
                                                                                  DESQUE
UNIT      DESQUE
NEXT      PPQUEUE
BACK      QUEUE
ERASE
JOIN      FBOXES
LINE      1209,1509
CALC      OR→1509
JOIN      DARROW
LINE      1257,1557
CALC      OR→1557
JOIN      DARROW
DELAY     .3
AT        1105
WRITE     ↑DELETE                                          ↑INSERT
DELAY     .2
AT        2006
WRITE     ↑FRONT      ↑SECOND      ↑THIRD      ↑FOURTH      ↑REAR
SIZE      2
AT        2827
WRITE     ↑QUEUE
SIZE
JOIN      PRESSEND
*
                                                                                  PPQUEUE
UNIT      PPQUEUE
ERASE
TERM      Q
BASE
INHIBIT   ERASE
CALC      OL→1
          OR→5
          N75→0
          N76→1500
JOIN      TRACK
JOIN      DOTDOT
JOIN      FRONT
JOIN      REAR
WRITE     ¬0AT,412¬1↑PRESS ↑,I↑, TO INSERT A NEW ITEM IN THE QUEUE.
          ↑PRESS ↑,D↑, TO DELETE AN ITEM FROM THE QUEUE.
```

```
1039  JOIN     SHOWN
1040  JUMP     QU1
1041  *                                                                  QU1

1042  UNIT     QU1
1043  BACK     DESQUE
1044  BASE     PRQUEUE
1045  INHIBIT  ERASE
1046  AT       931
1047  ERASE    3
1048  ARROW    931
1049  JOIN     NEXT
1050  LONG     1
1051  ANSWER
1052  CATCHUP
1053  JUMP     DEQUE
1054  MATCH    V1,I,D
1055  JUMP     V1,QU1,INS-Q,DEL-Q
1056  *                                                                  DOTDOT

1057  UNIT     DOTDOT
1058  CALC     N73→1
1059           N72→N76-18+7*OR
1060           OL→N76-11+7*OL                                            MV

1061  ENTRY    MV
1062  CALC     OL→OL+7
1063  JOIN     RECTAN
1064  JOIN     NUM
1065  CALC     OR→OL+104
1066  JOIN     N75=0,X,LARROW
1067  LINE     OL+104,OL+107
1068  CALC     OR→OL+107
1069           N73→N73+1
1070  JOIN     RARROW
1071  GOTO     OL>N72,X,MV
1072  CALC     OL→OL+7
1073  JOIN     RECTAN
1074  JOIN     NUM
1075  *                                                                  NUM

1076  UNIT     NUM
1077  CALCC    N73>9,N74→WHEREX+9,N74→WHEREX+13
1078  WRITE    ¬OAT,N74,WHEREY-24¬1¬OS,N73¬1
1079  *                                                                  TRACK

1080  UNIT     TRACK
1081  CALC     N2→OR-OL+1
1082           N41→N2          $$ NUMBER OF THE LAST NODE
1083           N31→1           $$ NUMBER OF THE FIRST NODE
1084           N40→OR          $$ POSITION OF THE REAR NODE
1085           N30→OL          $$ POSITION OF THE FRONT NODE
1086           N50→N2          $$ RANK OF THE LAST NODE INSERTED
1087  CALC     N5→OL
1088           N6→1                                                      DT

1089  ENTRY    DT
```

```
1090  CALC    N(50+N5)→N6
1091          N5→N5+1
1092          N6→N6+1
1093  GOTO    N5¬<OR,DT,X
1094  *
```

---------- MOVEQUEUE

SPACE LEFT = 15
INS-Q

```
1095  UNIT    INS-Q
1096  JOIN    CLEAN
1097  GOTO    N40-9,X,FULL
1098  JOIN    N75=1,CADEL,X
1099  CALC    N2→N2+1
1100          N50→N50+1
1101          N(50+N40+1)→N50
1102          N41→N50
1103  WRITE   ¬0AT,2822¬1↑PRESS ¬↑N↑E↑X↑T¬ TO INSERT
1104              THE NEW ITEM ¬0S,N41¬1
1105  JOIN    WAITN
1106  JOIN    N75=1,CADEF,X
1107  AT      2822
1108  ERASE   62
1109  MODE    ERASE
1110  GOTO    N75=1,LA2,X
1111  JOIN    N40=0,X,REAR
1112  JOIN    N2=2,FRONT1,X
1113  GOTO    LA3
                                                                LA2
1114  ENTRY   LA2
1115  GOTO    N40=0,LA3,X
1116  JOIN    N2=2,RIGHTF,RIGHTM
                                                                LA3
1117  ENTRY   LA3
1118  MODE    WRITE
1119  CALC    N40→N40+1
1120          OL→1496+N40≠7
1121          N73→N41
1122  GOTO    N75=1,LA1,X
1123  JOIN    N2-2,FRONT1,FRONT,X
1124  AT      1303
1125  WRITEC  N2=1¬,↑REAR¬,¬,
                                                                LA1
1126  ENTRY   LA1
1127  JOIN    N2=1$AND$N75=1,LEFTF,X
1128  GOTO    N40=1,LA0,X
1129  CALC    OR→1593+N40≠7
1130  JOIN    N75=1,LARROW,X
1131  DRAW    1593+N40≠7;1596+N40≠7
1132  CALC    OR→1596+N40≠7
1133  JOIN    RARROW
                                                                LA0
1134  ENTRY   LA0
1135  JOIN    RECTAN
1136  JOIN    NUM
1137  JOIN    N75=0,REAR,RIGHTM
1138  JOIN    SHOWN
```

```
*                                                                    DEL-Q

UNIT    DEL-Q
JOIN    CLEAN
GOTO    N2,X,EMPTY,X
JOIN    N75=1,CADEL,X
CALC    N2→N2-1
        N73→N31
WRITE   ¬0AT,2822¬1↑PRESS -↑N↑E↑X↑T- TO
         DELETE ITEM ¬0S,N31¬1
CALC    N31→N(50+N30+1)
JOIN    WAITN
JOIN    N75=1,CADEF,X
AT      2822
ERASE   62
MODE    ERASE
GOTO    N75=1,DG0,X
JOIN    N2=0,FRONT1,FRONT
JOIN    N2=0,REAR,X
GOTO    DG1                                                          DG0

ENTRY   DG0
JOIN    N2=0,RIGHTF,X
JOIN    LEFTM                                                        DG1

ENTRY   DG1
GOTO    N2=0,DEM,X
GOTO    N30=9,DEM,X
CALC    OR→1600+N30≠7
JOIN    N75=1,LARROW,X
DRAW    1600+N30≠7;1603+N30≠7
CALC    OR→1603+N30≠7
JOIN    RARROW                                                       DEM

ENTRY   DEM
CALC    OL→1496+N30≠7
JOIN    RECTAN
JOIN    NUM
MODE    WRITE
CALC    N30→N30+1
GOTO    N75=1,DG2,X
JOIN    N2-1,X,FRONT1,FRONT
GOTO    DG3                                                          DG2

ENTRY   DG2
JOIN    N2¬=0,LEFTM,X                                                DG3

ENTRY   DG3
JOIN    SHOWN
JOIN    N2=0,RESET,X
JUMP    N75=0,DU1,DU1
*                                                                    DEQUE

UNIT    DEQUE
NEXT    DESDEQUE
BACK    QUEUE
```

```
BASE
ERASE
SIZE    2
AT      602
WRITE   A DEQUE↑!
SIZE
AT      803
WRITE   (DOUBLE ENDED QUEUE) IS A LINEAR LIST FOR WHICH ALL INSERTIONS
        AND DELETIONS (AND USUALLY ALL ACCESSES) ARE MADE AT THE ENDS
        OF THE LIST.
DELAY   .4
JOIN    WAIT
GOTO    KEY=BACK,DEQUE,X
AT      1811
WRITE   * ↑A DEQUE IS MORE GENERAL THAN A STACK
             OR A QUEUE.


        * ↑WHEN REFERING TO DEQUES, WE SPEAK OF
             THE LEFT AND RIGHT ENDS.
JOIN    PRESSEND
*                                                                    RECTAN
UNIT    RECTAN
DRAW    OL!OL+4!OL+204!OL+200!OL
*
-------------------------------------------------------------------------------- DEQUE
                                                                    SPACE LEFT = 23
                                                                       DESDEQUE
UNIT    DESDEQUE
NEXT    PPDEQUE
BACK    DEQUE
ERASE
JOIN    FIGDEQUE
DELAY   .2
AT      1204
WRITE   ↑LEFTMOST      ↑SECOND                ↑SECOND       ↑RIGHTMOST
AT      1301
WRITE                   FROM LEFT             FROM RIGHT
AT      2203
WRITE   ↑INSERT OR DELETE                            ↑INSERT OR DELETE
SIZE    2
AT      525
WRITE   ↑DEQUE
SIZE    0
JOIN    PRESSEND
*                                                                    FIGDEQUE
UNIT    FIGDEQUE
JOIN    FBOXES
CALC    OR→1613
JOIN    LARROW
CALC    OR→1625
JOIN    LARROW
CALC    OR→1637
JOIN    LARROW
```

```
1241  JOIN    LARROW
1242  LINE    1709,2009
1243  CALC    OR+1709
1244  JOIN    UARROW
1245  LINE    1757,2157
1246  CALC    OR+1757
1247  JOIN    UARROW
1248  *                                                          CLEAN

1249  UNIT    CLEAN
1250  INHIBIT ERASE
1251  AT      731
1252  ERASE   62
1253  AT      931
1254  ERASE   3
1255  MODE    ERASE
1256  GOTO    N75=1,CL1,X
1257  JOIN    NEXT
1258  GOTO    CL2
                                                             CL1
1259  ENTRY   CL1
1260  AT      3130
1261  ERASE   3
1262  JOIN    V1=0,INSERTT,DELETEE
                                                             CL2
1263  ENTRY   CL2
1264  MODE    WRITE
1265  *                                                          RESET

1266  UNIT    RESET
1267  WRITE   ¬0AT,2102¬1+INSERT
1268  JOIN    N75=1,OR,X
1269  WRITE   ¬0AT,2302¬1+DELETE
1270  DRAW    19051705
1271  CALC    OR+1705
1272          N40+0
1273          N30+1
1274          N31+N50+1
1275          N41+N50+1
1276  *
1277  JOIN    UARROW                                             LUARROW

1278  UNIT    LUARROW
1279  DRAW    1898+N30#7:1698+N30#7
1280  CALC    OR+1698+N30#7
1281  JOIN    UARROW
1282  *                                                          PPDEQUE

1283  UNIT    PPDEQUE
1284  ERASE
1285  TERM    E
1286  BASE
1287  INHIBIT ERASE
1288  CALC    OL+4
1289          OR+6
1290          N75+1
```

```
1291          N76→1500
1292  JOIN    TRACK
1293  JOIN    DOTDOT
1294  CALC    V3→0
1295  JOIN    LEFTM
1296  CALC    V3→1
1297  JOIN    RIGHTM
1298  WRITE   ¬@AT,412¬1↑PRESS ↑,I↑, TO INSERT A NEW ITEM IN THE DEQUE
1299          ↑PRESS ↑,D↑, TO DELETE AN ITEM FROM THE DEQUE.
1300  JUMP    DU1
1301  *                                                                    DU1

1302  UNIT    DU1
1303  BACK    DESDEQUE
1304  BASE    PPDEQUE
1305  INHIBIT ERASE
1306  AT      931
1307  ERASE   3
1308  ARROW   931
1309  LONG    1
1310  JOIN    NEXT
1311  JOIN    SHOWN
1312  ANSWER
1313  CATCHUP
1314  PACK    N100,DRILLC
1315  JUMP    TEST,ALLOC,YY,YY                  $$  DRILLC(STR2)
1316  MATCH   V1,I,D
1317  JUMP    V1,DU1,SIDE
1318  *                                                                    SIDE

1319  UNIT    SIDE
1320  INHIBIT ERASE
1321  JUMP    N2=9$AND$V1=0,FULL,X
1322  JUMP    N2=0$AND$V1=1,EMPTY,X
1323  AT      731
1324  ERASE   62
1325  MODE    ERASE
1326  JOIN    NEXT
1327  MODE    WRITE
1328  AT      931
1329  ERASE   3
1330  JOIN    V1=0,INSERTT,DELETEE                                         ES

1331  ENTRY   ES
1332  INHIBIT ERASE
1333  AT      3130
1334  ERASE   3
1335  ARROW   3130
1336  LONG    1
1337  ANSWER
1338  CATCHUP
1339  JUMP    ES
1340  MATCH   V3,L,R
1341  JUMP    V3=0$AND$V1=0,INSLEFT,X
1342  JUMP    V3=0$AND$V1=1,DEL-Q,X
1343  JUMP    V3=1$AND$V1=0,INS-Q,X
1344  JUMP    V3=1$AND$V1=1,DELRIGHT,ES
```

```
1345  *
--------------------------------------------------------------------------------------------------------------------------------------- MOVEDEQUE
                                                                                                                  SPACE LEFT = 13
                                                                                                                     INSLEFT
1346  UNIT    INSLEFT
1347  JOIN    CLEAN
1348  GOTO    N30=1$AND$N2¬=0,FULL,X
1349  JOIN    CADEL
1350  CALC    N2←N2+1
1351          N50←N50+1
1352          N31←N50
1353          N(50+N30-1)←N50
1354          N73←N31
1355  WRITE   ¬0AT,2822¬1↑PRESS =↑N↑E↑X↑T= TO INSERT
1356              THE NEW ITEM ¬0S,N31¬1
1357  JOIN    WAITN
1358  JOIN    CADEF
1359  AT      2822
1360  ERASE   62
1361  MODE    ERASE
1362  JOIN    N2=2,X,LEFTF,LEFTM
1363  MODE    WRITE
1364  CALCC   N30=1,N40←N40+1,N40←N40
1365  CALCC   N30=1,N30←N30,N30←N30-1
1366  CALC    OL←1496+N30*7
1367  GOTO    N30=1$AND$N2=1,ILO,X
1368  CALC    OR←1603+N30*7
1369  JOIN    RARROW
1370  DRAW    1603+N30*7:1600+N30*7
1371  CALC    OR←1600+N30*7
1372  JOIN    LARROW
                                                                                                                  ILO
1373  ENTRY   ILO
1374  JOIN    RECTAN
1375  JOIN    NUM
1376  CALC    N5←N40
1377          N40←N30
1378  JOIN    N2=1,LEFTF,LEFTM
1379  CALC    N40←N5
1380  JOIN    N2=1,RIGHTF,X
1381  JOIN    SHO←N
1382  JUMP    DU1
1383  *
                                                                                                                  DELRIGHT
1384  UNIT    DELRIGHT
1385  JOIN    CLEAN
1386  GOTO    N2,EMPTY,X
1387  JOIN    CADEL
1388  CALC    N2←N2-1
1389  WRITE   ¬0AT,2822¬1↑PRESS =↑N↑E↑X↑T= TO
1390              DELETE ITEM ¬0S,N41¬1
1391  JOIN    WAITN
1392  JOIN    CADEF
1393  AT      2822
1394  ERASE   62
1395  MODE    ERASE
```

```
1396  JOIN    N2=0,LEFTF,X
1397  JOIN    RIGHTM
1398  GOTO    N40=1,DR0,X
1399  CALC    OR←1593+N40*7
1400  JOIN    LARROW
1401  DRAW    1593+N40*7:1696+N40*7
1402  CALC    OR←1596+N40*7
1403  JOIN    RARROW
                                                                    DR0
1404  ENTRY   DR0
1405  CALC    OL←1496+N40*7
1406          N73←N41
1407  JOIN    RECTAN
1408  JOIN    NUM
1409  MODE    WRITE
1410  CALC    N40←N40-1
1411          N41←N(50+N40)
1412  JOIN    N2¬=0,RIGHTM,X
1413  JOIN    SHOWN
1414  JOIN    N2=0,RESET,X
1415  JUMP    DU1
1416  *
                                                                    OR
1417  UNIT    OR
1418  WRITE   ¬0AT,2204¬1OR
1419  *
                                                                    RIGHTM
1420  UNIT    RIGHTM
1421  CALCC   N40=9,N5←1356,N5←1294+N40*7
1422  WRITE   ¬0AT,N5¬1↑RIGHTMOST
1423  WRITE   ¬0AT,2095+N40*7¬1↑INSERT
1424            OR
1425          ↑DELETE
1426  DRAW    1698+N40*7:1698+N40*7
1427  CALC    OR←1698+N40*7
1428  JOIN    UARROW
1429  *
                                                                    RIGHTF
1430  UNIT    RIGHTF
1431  CALCC   N40=9,N5←1356,N5←1294+N40*7
1432  WRITE   ¬0AT,N5¬1↑RIGHTMOST
1433  *
                                                                    LEFTM
1434  UNIT    LEFTM
1435  WRITE   ¬0AT,1094+N30*7¬1↑LEFTMOST
1436  WRITE   ¬0AT,2095+N30*7¬1↑INSERT
1437            OR
1438          ↑DELETE
1439  JOIN    LUARRO*
1440  *
                                                                    LEFTF
1441  UNIT    LEFTF
1442  WRITE   ¬0AT,1094+N30*7¬1↑LEFTMOST
1443  *
                                                                    INSERTT
1444  UNIT    INSERTT
1445  WRITE   ¬0AT,2818¬1↑PRESS -L- TO INSERT ON THE LEFT.
```

```
1446          +PRESS -R- TO INSERT ON THE RIGHT.
1447  *
                                                                                                                  DELETEE
1448  UNIT    DELETEE
1449  WRITE   ¬0AT,2818¬1+PRESS -L- TO DELETE ON THE LEFT,
1450          +PRESS -R- TO DELETE ON THE RIGHT.
1451  *
                                                                                                                  CADEL
1452  UNIT    CADEL
1453  CALCC   N40=0,+6*1,N6*N40
1454  CALCC   V3=0,N5*N30,N5*N6
1455  CALC    N7*56*N5-56
1456          N8*32*V1
1457  DRAW    4+N7,192-N8!58+N7,192-N8!58+N7,176-N8!4+N7,176-N8!4+N7,192-N8
1458  *
                                                                                                                  CADEF
1459  UNIT    CADEF
1460  MODE    ERASE
1461  DRAW    4+N7,192-N8!58+N7,192-N8!58+N7,176-N8!4+N7,176-N8!4+N7,192-N8
1462  MODE    WRITE
1463  *
 ----------------------------------------------------------------------------------------------------------------- MOVEDEQUE
                                                                                                  SPACE LEFT = 199
                                                                                                        EMPTY
1464  UNIT    EMPTY
1465  INHIBIT ERASE
1466  AT      731
1467  WRITEC  N75=0¬,+SORRY, THE QUEUE IS EMPTY.¬,+SORRY, THE DEQUE IS EMPTY.¬,
1468  JUMP    N75=0,QU1,QU1
1469  *
                                                                                                                  FULL
1470  UNIT    FULL
1471  INHIBIT ERASE
1472  AT      731
1473  GOTO    N75=1,EM0,X
1474  WRITE     +SORRY, THE QUEUE IS FULL
1475          (NO MORE ROOM ON THE SCREEN)
1476  JUMP    QU1
                                                                                                        EM0
1477  ENTRY   EM0
1478  GOTO    N2=9,X,EM2
1479  WRITE     +SORRY, THE DEQUE IS FULL
1480          (NO MORE ROOM ON THE SCREEN)
1481  JUMP    QU1
                                                                                                        EM2
1482  ENTRY   EM2
1483  WRITEC  V3=0¬,+SORRY, NO MORE ROOM ON THE LEFT.¬,+SORRY, NO MORE ROOM ON THE RIGHT.¬,
1484  JUMP    QU1
1485  *
                                                                                                                  REAR
1486  UNIT    REAR
1487  WRITE   ¬0AT,1296+N40*7¬1+REAR
1488  WRITE   ¬0AT,2095+N40*7¬1+INSERT
1489  JOIN    RUARROW
1490  *
                                                                                                                  RUARROW
```

```
UNIT     RUARROW
DRAW     1898+N40#7;1698+N40#7
CALC     OR+1698+N40#7
JOIN     UARROW
*
                                                                                FRONT
UNIT     FRONT
CALC     N5+1296+N30#7
CALC     N6+2095+N30#7
JOIN     FRONTT
*
                                                                                FRONT1
UNIT     FRONT1
CALC     N5+1096+N30#7
CALC     N6+2295+N30#7
JOIN     FRONTT
*
                                                                                FRONTT
UNIT     FRONTT
WRITE    ¬0AT,N5¬1↑FRONT
WRITE    ¬0AT,N6¬1↑DELETE
JOIN     LUARROW
*
-------------------------------------------------------------------------------------------------------------------------------- ALLOCATION
                                                                                SPACE LEFT = 89
                                                                                LINK4
UNIT     LINK4
FIGURE   1639,1652,1952,1939,2239,2252,2552,2539,2839,2852,1352,1339,2839,2847,1347
*
                                                                                LINK5
UNIT     LINK5
AT       1431
WRITE    ↑ADDRESS
            ↑A      ITEM 1    ↑B
AT       1834
WRITE    ↑B      ITEM 2    ↑C
AT       2134
WRITE    ↑C      ITEM 3    ↑D
AT       2434
WRITE    ↑D      ITEM 4    ↑E
AT       2734
WRITE    ↑E      ITEM 5   ≥JI
*
                                                                                AL
UNIT     AL
JOIN     WAIT
GOTO     KEY=BACK,ALLOC,X
*
                                                                                ALLOC
UNIT     ALLOC
INHIBIT  ERASE
BASE
ERASE
SIZE     2
AT       203
WRITE    ↑STORAGE ALLOCATION↑!
```

```
1530  SIZE
1531  AT      403
1532  WRITE      ↑TWO NOTATIONS ARE USED IN THE FOLLOWING DISCUSSION↑!
1533
1534                 ↑L↑O↑C      ↑L↑O↑C[↑V] DENOTES THE MEMORY LOCATION OF ↑V.
1535                 ↑V ←↑A ↑E   GIVES VARIABLE ↑V THE VALUE OF EXPRESSION ↑E.
1536  JOIN    AL
1537  AT      916
1538  WRITE   ↑THERE ARE TWO MAIN WAYS OF KEEPING
1539          A LINEAR LIST IN A COMPUTER MEMORY↑!
1540  DELAY   .5
1541  AT      1204
1542  WRITE      ↑SEQUENTIAL ALLOCATION          ↑LINKED ALLOCATION
1543  JOIN    AL
1544  FIGURE  1313,1613,1621,1921,1913,2213,2221,2521,2513,2813,2821,1321,1313,2813
1545  JOIN    AL
1546  AT      1404
1547  WRITE    ↑ADDRESS
1548          ↑Lv↑≠≤v↑Ξ↑C       ITEM 1
1549  AT      1804
1550  WRITE   ↑Lv↑≠≤v↑Ξ↑2C      ITEM 2
1551  AT      2104
1552  WRITE   ↑Lv↑≠≤v↑Ξ↑3C      ITEM 3
1553  AT      2404
1554  WRITE   ↑Lv↑≠≤v↑Ξ↑4C      ITEM 4
1555  AT      2704
1556  WRITE   ↑Lv↑≠≤v↑Ξ↑5C      ITEM 5
1557  WRITE   ¬0AT,2911¬1   C IS THE NUMBER OF WORDS IN EACH NODE
1558           ↑Lv↑≠≤v↑Ξ IS THE BASE ADDRESS OF THE FIRST NODE
1559  JOIN    AL
1560  JOIN    LINK4
1561  DELAY   .4
1562  JOIN    AL
1563  JOIN    LINK5
1564  JOIN    AL
1565  AT      403
1566  ERASE   420
1567  FIGURE  309,357,1057,1009,309
1568  AT      520
1569  WRITE   1 - ↑SEQUENTIAL ALLOCATION.
1570          2 - ↑LINKED ALLOCATION.
1571  AT      78,384
1572  WRITE   ↑PRESS 1 OR 2 TO SEE ONE OF THESE TWO SECTIONS
                                                                          ALLOC2
1573  ENTRY   ALLOC2
1574  BACK    LLIST
1575  INHIBIT ERASE
1576  AT      932
1577  ERASE   3
1578  ARROW   932
1579  LONG    1
1580  MATCH   V1,1,2
1581  JUMP    V1,INSIST,SEQUEN,LINKED
1582  *
```

---------- SEQUENTIAL

SPACE LEFT = 60

```
                                                                                INSIST
UNIT     INSIST
INHIBIT  ERASE
FIGURE   710,723,823,810,710
JUMP     ALLOC2
*
                                                                                SEQUEN
UNIT     SEQUEN
NEXT     SEQ2
BACK     ALLOC
BASE
ERASE
SIZE     1
AT       405
WRITE    SEQUENTIAL ALLOCATION
SIZE
FIGURE   203,228,528,503,203
JOIN     WAIT
GOTO     KEY=BACK,SEQUEN,X
CATCHUP
AT       902
WRITE       ↑THE SIMPLEST AND MOST NATURAL WAY TO KEEP A LINEAR LIST
         INSIDE A COMPUTER IS TO PUT THE LIST ITEMS IN SEQUENTIAL
         LOCATIONS, ONE NODE AFTER THE OTHER.
JOIN     WAIT
GOTO     KEY=BACK,SEQUEN,X
AT       1207
WRITE    ↑WE WILL HAVE↑:

              ↑L↑O↑C(↑X[J+1])= ↑L↑O↑C(↑X[J])+ C,

         WHERE C IS THE NUMBER OF WORDS BY NODE.
         ↑IN GENERAL↑:

              ↑L↑O↑C(↑X[J])= ↑Lv↑≠≤v↑≡+ CJ,

         WHERE ↑Lv↑≠≤v↑≡ IS THE BASE ADDRESS LOCATION OF AN ARTIFICIALLY
         ASSUMED NODE ↑X[0].
DELAY    .4
JOIN     PRESSEND
*
                                                                                AS
UNIT     AS
JOIN     WAIT
GOTO     KEY=BACK,SEQ2,X
*
                                                                                SEQ2
UNIT     SEQ2
NEXT     SEQ21
BACK     SEQUEN
ERASE
AT       202
WRITE    ↑SEQUENTIAL ALLOCATION IS QUITE CONVENIENT FOR A STACK
         ↑WE NEED ONLY VARIABLE ↑T CALLED THE STACK POINTER.
         ↑WHEN THE STACK IS EMPTY ↑T=0.
JOIN     AS
AT       602
```

```
WRITE        ↑TO PLACE A NEW ELEMENT ON THE TOP OF THE STACK, WE SET↑!

                        ↑T ← ↑T+1!
                        ↑X[↑T] ← ↑Y.
JOIN    AS
AT      1102
WRITE        ↑TO DELETE A NODE FROM THE STACK(WHEN IT IS NOT EMPTY),
             WE SET↑!

                        ↑Y ← ↑X[↑T]!
                        ↑T  ← ↑T-1.
DELAY   .2
JOIN    AS
AT      1702
WRITE   ↑FOR A QUEUE AND A DEQUE WE NEED AT LEAST TWO POINTERS
        ↑F AND ↑R ( FOR THE FRONT AND THE REAR OF THE QUEUE)↑!
                 WHEN  ↑F= ↑R= 0 THE QUEUE IS EMPTY,
JOIN    AS
AT      2102
WRITE        ↑TO INSERT AN ELEMENT AT THE REAR OF THE QUEUE, WE SET↑!

                        ↑R ← ↑R+1!
                        ↑X[↑R] ← ↑Y.
JOIN    AS
AT      2602
WRITE        ↑TO REMOVE THE FRONT NODE, WE SET↑!

                        ↑F ← ↑F+1!
                        ↑Y ← ↑X[↑F]!
                        IF ↑F= ↑R THEN SETS ↑F ← ↑R ← 0.
DELAY   .2
JOIN    PRESSEND
*
```

---------- AVAIL LIST

SPACE LEFT = 7

SEQ21

```
UNIT    SEQ21
PACK    N100,DRILLD
NEXT    TEST,AVAIL,YY,YY              $$ DRILLD(STR2)
BACK    SEQ2
ERASE
AT      903
WRITE      ↑IN THIS WAY THERE IS TERRIBLY WASTEFUL OF STORAGE SPACE.
        ↑THE PREVIOUS METHOD IS ONLY USED WHEN ↑F IS KNOWN TO CATCH
        UP TO ↑R QUITE REGULARLY.
JOIN    WAIT
GOTO    KEY=BACK,SEQ21,X
AT      1403
WRITE      ↑OTHERWISE, WE CAN SET ASIDE ↑M NODE ↑X[1],... ↑X[↑M]
        ARRANGED IMPLICITLY IN CIRCLE WITH ↑X[1] FOLLOWING ↑X[↑M].

           ↑THEN THE INSERTION AND DELETION PROCESSES BECOME↑!
JOIN    WAIT
GOTO    KEY=BACK,SEQ21,X
AT      1903
WRITE   ↑I↑U↑S↑E↑R↑T↑!
```

```
                IF ↑R= 0 THEN  ↑R ← 1 ;
                OTHERWISE  ↑R ← ↑R+1;   ↑X[↑R] ← ↑Y.
JOIN     WAIT
GOTO     KEY=BACK.SEQ21.X
AT       2303
WRITE    ↑D↑E↑L↑E↑T↑E↑;
                IF ↑F= ↑M THEN  ↑F ← 1 ;
                OTHERWISE  ↑F ← ↑F+1;   ↑Y ← ↑X[↑F].
JOIN     PRESSEND
*

UNIT     AVAIL
NEXT     AVAIL2
BACK     LINK3
BASE
INHIBIT ERASE
ERASE
AT       205
WRITE    ↑THE USE OF LINKED ALLOCATION IMPLIES THE EXISTENCE
         OF SOME MECHANISM FOR FINDING EMPTY SPACE AVAILABLE
         FOR A NEW NODE, WHEN WE WISH TO INSERT A NEW NODE
         ONTO A LIST.
JOIN     AVA
AT       605
WRITE    ↑THIS IS USUALLY DONE BY HAVING A SPECIAL LIST CALLED
SIZE     1
AT       818
WRITE    LIST OF AVAILABLE SPACE
SIZE     0
FIGURE   717,743,843,817,717
DELAY    .3
AT       905
WRITE            (↑WE WILL CALL IT THE  ↑A↑V↑A↑I↑L$   LIST.)

         ↑THUS,IF WE WANT TO SET LINK ↑X TO THE ADDRESS OF A NEW
         NODE,AND TO RESERVE THAT NEW NODE FOR FUTURE USE,WE
         CAN PROCEED AS FOLLOW↑;

                  ↑X ← ↑A↑V↑A↑I↑L.   ↑A↑V↑A↑I↑L ← ↑L↑I↑N↑K[↑A↑V↑A↑I↑L].
JOIN     AVA
AT       1905
WRITE    ↑THIS REMOVES THE TOP OF THE ↑A↑V↑A↑I↑L STACK AND MAKES ↑X
         POINT TO THE NODE JUST REMOVED.
JOIN     AVA
AT       2305
WRITE    ↑WHEN A NODE IS DELETED, THE PROCESS CAN BE REVERSED↑;

                  ↑L↑I↑N↑K[↑X] ← ↑A↑V↑A↑I↑L.   ↑A↑V↑A↑I↑L ← ↑X.
JOIN     AVA
AT       2805
WRITE    ↑THIS OPERATION PUTS THE NODE ADDRESSED BY ↑X$  BACK ONTO
         THE ↑A↑V↑A↑I↑L LIST.
DELAY    .3
JOIN     AVA
AT       1205
```

```
1753  ERASE   530
1754  FIGURE  1104,1160,2160,2104,1104
1755  AT      1205
1756  WRITE   ↑WE HAVE OMITTED TO TEST FOR OVERFLOW↑!
1757
1758          WE DIDN↑T CHECK IF ALL AVAILABLE MEMORY SPACE HAS BEEN
1759          TAKEN, SO THE OPERATION HAS TO BE DEFINED AS FOLLOWS↑!
1760
1761
1762             IF ↑A↑V↑A↑I↑L = ≥JI≥, THEN ↑O↑V↑E↑R↑F↑L↑O↑W,
1763                              OTHERWISE  ↑X ← ↑A↑V↑A↑I↑L
1764                                         ↑A↑V↑A↑I↑L ← ↑L↑I↑N↑K[↑A↑V↑A↑I↑L].
1765  DRAW    1706;1758;2058;2006;1706
```

INSERTION SPACE LEFT = 50

```
1766  DELAY   .4
1767  JOIN    AVA
1768  MOVE    REWRITE
1769  AT      1205
1770  WRITE   ↑THIS OPERATION REMOVE THE TOP OF THE ↑A↑V↑A↑I↑L LIST AND
1771          RESERVE THAT NODE FOR FUTURE USE.
1772          ↑THIS OPERATION WILL BE DENOTED↑!
1773
1774                                ↑X ≥MNN≥ ↑A↑V↑A↑I↑L
1775  DRAW    1524;1537;1637;1624;1524
1776  JOIN    AVA
1777  AT      2305
1778  ERASE   420
1779  AT      3017
1780  WRITE   ↑L↑I↑N↑K[↑X] ← ↑A↑V↑A↑I↑L,   ↑A↑V↑A↑I↑L ← ↑X.
1781  FIGURE  2914,2948,3048,3014,2914
1782  AT      2405
1783  WRITE   ↑THIS OPERATION PUTS THE NODE ADDRESSED BY ↑X BACK ONTO
1784          THE ↑A↑V↑A↑I↑L LIST.
1785  JOIN    AVA
1786  AT      2605
1787  WRITE   ↑THIS OPERATION WILL BE DENOTED↑!
1788
1789                                ↑A↑V↑A↑I↑L ≥MNN ≥↑X
1790  DRAW    2724;2737;2837;2824;2724
1791  DRAW    2304;2360;3160;3104;2304
1792  JOIN    PRESSEND
1793  *                                                                    AVA

1794  UNIT    AVA
1795  JOIN    WAIT
1796  GOTO    KEY=BACK,AVAIL,X
1797  *                                                                    AVB

1798  UNIT    AVB
1799  JOIN    WAIT
1800  GOTO    KEY=BACK,AVAIL2,X
1801  *                                                                    STEP

1802  UNIT    STEP
1803  WRITE   ¬0AT,3225¬1 ↑PRESS ←↑N↑E↑X↑T→ TO DO STEP
```

```
*
UNIT     AVAIL2
NEXT     LINKED
BACK     SEQUEN
INHIBIT ERASE
ERASE
AT       630
WRITE    *A STACK IS THE SIMPLEST KIND
         OF LINKED LIST.*THE FOLLOWING
         FIGURE SHOWS A STACK, WITH A
         POINTER *T TO THE TOP OF THE
         STACK.
DELAY    .3
JOIN     AVA
JOIN     FIGSTACK
AT       613
WRITE    *T
LINE     614,620
CALC     OR*620
JOIN     RARROW
AT       160,412
WRITE    INFOV5
CIRCLE   2,209,417,15,360
FIGURE   627,927,822,1022
CALC     OR*1022
JOIN     DARROW
AT       160,332
WRITE    INFOV4
CIRCLE   2,209,337,15,360
FIGURE   1127,1327,1322,1522
CALC     OR*1522
JOIN     DARROW
AT       160,252
WRITE    INFOV3
CIRCLE   2,209,257,15,360
FIGURE   1627,1827,1822,2022
CALC     OR*2022
JOIN     DARROW
AT       160,172
WRITE    INFOV2
CIRCLE   2,209,177,15,360
FIGURE   2127,2327,2322,2522
CALC     OR*2522
JOIN     DARROW
AT       160,92
WRITE    INFOV1
CIRCLE   2,209,97,15,360
LINE     2627,3027
WRITE    ¬UAT,3026¬12KL
AT       1330
WRITE    *WHEN THE STACK IS EMPTY, THE
         POINTER WILL HAVE THE VALUE ≥JI≥.
JOIN     AVA
MODE     REWRITE
WRITE    ¬UAT,630¬1*TO INSERT (PUSH DOWN) THE
         INFORMATION *Y ONTO THE TOP
```

```
          OF THE STACK WE NEED AN
          AUXILIARY POINTER VARIABLE ↑P↑I

--------------------------------------------------------------------------------- DELETION
                                                                  SPACE LEFT = 4

                    A=     ↑P ≥MNN 2↑A↑V↑A↑I↑L.
                    B=     ↑I↑N↑F↑O[↑P] → ↑Y.
                    C=     ↑L↑I↑N↑K[↑P] → ↑T.
                    D=     ↑T → ↑P.
AT       213
WRITE    ↑P
LINE     214,220
CALC     OR→220
JOIN     HARROW
MODE     WRITE
JOIN     STEP
WRITE    ¬UAT,3249¬1 A
JOIN     AVB
AT       1134
ERASE    3
CALC     OR→120
JOIN     NODE
DELAY    .5
JOIN     STEP
WRITE    ¬UAT,3249¬1 B
JOIN     AVB
AT       1234
ERASE    3
AT       184,472
WRITE    ↑Y
DELAY    .5
JOIN     STEP
WRITE    ¬UAT,3249¬1 C
JOIN     AVB
AT       1334
ERASE    3
CIRCLE   2,209,481,15,360
DRAW     227;427;414;614
DELAY    .4
JOIN     STEP
WRITE    ¬UAT,3249¬1 D
JOIN     AVB
AT       1434
ERASE    3
AT       613
ERASE    1
AT       213
ERASE    1
AT       213
WRITE    ↑T
DRAW     1038;1054;1454;1438;1038
JOIN     AVB
AT       113
ERASE    262
LINE     520,528
AT       613
```

```
WRITE   ↑T
AT      1730
WRITE   ↑TO SET ↑Y TO THE INFORMATION
        AT THE TOP OF THE STACK AND
        SO POP UP THE STACK WE NEED
        AN AUXILIARY POINTER ↑P↑!

             IF ↑T = ≥JI≥, THEN OVERFLOW.

             OTHERWISE  A-  ↑P ↗ ↑T,
                        B-  ↑T ↗ ↑L↑I↑N↑K[↑P],
                        C-  ↑Y ↗ ↑I↑N↑F↑O[↑P],
                        D-  ↑A↑V↑A↑I↑L ≥MNN≥ ↑P.
JOIN    AVB
AT      413
WRITE   ↑P
LINE    414,420
CALC    OR↗420
JOIN    RARROW
DELAY   .3
JOIN    STEP
WRITE   ¬0AT,3249¬1 A
JOIN    AVB
AT      2446
ERASE   2
AT      419
ERASE   1
AT      519
ERASE   1
FIGURE  414,424,524
CALC    OR↗524
JOIN    DARROW
DELAY   .3
JOIN    STEP
WRITE   ¬0AT,3249¬1 B
JOIN    AVB
AT      2546
ERASE   2
AT      613
ERASE   7
AT      719
ERASE   1
LINE    520,720
AT      1113
WRITE   ↑T
LINE    1114,1120
CALC    OR↗1120
JOIN    RARROW
DELAY   .4
JOIN    STEP
WRITE   ¬0AT,3249¬1 C
JOIN    AVB
AT      2646
ERASE   2
AT      319
WRITE   ↑Y ↗ INFOVS
DELAY   .4
```

```
1971  JOIN     STEP
1972  WRITE    ¬0AT,3249¬1 D
1973  JOIN     AVR
1974  AT       2746
1975  ERASE    2
1976  AT       413
1977  ERASE    11
1978  AT       520
1979  ERASE    9
1980  AT       620
1981  ERASE    9
1982  AT       720
1983  ERASE    9
1984  AT       822
1985  ERASE    6
1986  AT       922
1987  ERASE    1
1988  AT       1021
1989  ERASE    2
1990  LINE     1021,1023
1991  LINE     1020,1028
1992  DELAY    .2
1993  FIGURE   2347,2363,2763,2747,2347
1994  JOIN     PRESSED
1995  *
```

---------- LINKED

SPACE LEFT = 96

LP

```
1996  UNIT     LP
1997  JOIN     WAIT
1998  GOTO     KEY=BACK,LINKED,X
1999  *
```

LINKED

```
2000  UNIT     LINKED
2001  NEXT     LINK3
2002  BACK     AVAIL
2003  BASE
2004  ERASE
2005  SIZE     1
2006  AT       305
2007  WRITE    LINKED ALLOCATION
2008  SIZE     0
2009  FIGURE   104,123,423,404,104
2010  CATCHUP
2011  JOIN     LP
2012  AT       708
2013  WRITE    *INSTEAD OF KEEPING A LINEAR LIST IN SEQUENTIAL
2014           MEMORY LOCATIONS WE CAN USE ANOTHER SCHEME IN
2015           *HICH EACH NODE CONTAINS A LINK TO THE NEXT
2016           NODE OF THE LIST.
2017  JOIN     LP
2018  JOIN     LINK4
2019  DELAY    .4
2020  JOIN     LP
2021  JOIN     LINK5
2022  JOIN     LP
```

```
AT        18,8
WRITE     ↑LINKS ARE OFTEN SHOWN BY
          ARROWS AND THE ACTUAL
          MEMORY LOCATIONS ARE
          USUALLY IRRELEVANT.
DELAY     .3
JOIN      LP
MODE      ERASE
JOIN      LINK4
JOIN      LINK5
MODE      WRITE
JOIN      LP
JOIN      DISPLAY2
DELAY     .3
JOIN      LP
MODE      REWRITE
AT        160,8
WRITE     ↑LINKED ALLOCATION TAKES UP ADDITIONAL MEMORY SPACE
          FOR THE LINKS.

          ↑REFERENCES TO RANDOM PARTS OF THE LIST ARE MUCH
          FASTER IN THE SEQUENTIAL CASE.


          ↑B↑U↑T


          ↑THE LINKED SCHEME MAKES IT EASIER TO JOIN TWO LISTS
          TOGETHER OR TO BREAK ONE APART.

          ↑THE LINKED SCHEME LENDS ITSELF TO MORE COMPLEX
          STRUCTURE THAN SIMPLE LINEAR LISTS.
JOIN      PRESSEND
*
```

LINK1

```
UNIT      LINK1
JOIN      FBOXES
AT        130,1
WRITE     ↑F↑I↑R↑S↑T
FIGURE    1403,1603,1605
CALC      OK←1605
JOIN      RARROW
AT        35,249
WRITE     ITEM 1
CIRCLE    2,88,256
LINE      1612,1613
AT        131,249
WRITE     ITEM 2
CIRCLE    2,184,256
LINE      1624,1625
AT        227,249
WRITE     ITEM 3
CIRCLE    2,280,256
LINE      1636,1637
AT        323,249
WRITE     ITEM 4
CIRCLE    2,376,256
```

```
LINE     1648,1649
AT       419,249
WRITE    ITEM 5
```

---------- DELETE
SPACE LEFT = 50

```
CIRCLE   2,472,256
LINE     1660,1661
FIGURE   1661,1663,1863
WRITE    ¬0AT,1862¬12KL
*                                                                                DELINS

UNIT     DELINS
AT       118,160
WRITE    *TYPE *,INSERT*, OR *,DELETE*, TO INSERT
                 OR DELETE A NEW ITEM.
                   (THEN PRESS NEXT)
*                                                                                LINK3

UNIT     LINK3
INHIBIT  ERASE
ERASE
CALC     V31←-1
         V32←-1
         V33←-1
         V34←-1
         V35←-1
         V41←0
         V42←0
         V43←0
         V44←0
         V45←0
AT       703
WRITE    *≠≤*≡ *IT IS EASY TO DELETE AN ITEM FROM WITHIN A LINKED LIST,
           WE NEED TO CHANGE ONLY ONE LINK.

         *≠≤*≡ *IT IS EASY TO INSERT AN ITEM INTO THE LIST, WE NEED
           CHANGE ONLY TWO LINKS.
JOIN     LINK1                                                                   J2

ENTRY    J2
INHIBIT  ERASE
JOIN     DEPRESS                                                                 J21

ENTRY    J21
INHIBIT  ERASE
JOIN     DELINS
JOIN     NEXT                                                                    J22

ENTRY    J22
INHIBIT  ERASE
BACK     AVAIL
AT       2728
ERASE    8
ARROW    2728
LONG     6
ANSWER
```

```
2128  JUMP     DISPA
2129  MATCH    V1,INSERT,DELETE
2130  JUMP     V1,MISS,INSERT,DELETE
2131  *
                                                                    INSERT
2132  UNIT     INSERT
2133  INHIBIT  ERASE
2134  MODE     ERASE
2135  FIGURE   2115,2141,2241,2215,2115
2136  JOIN     DELIMS
2137  AT       2728
2138  ERASE    8
2139  JOIN     NEXT
2140  MODE     WRITE
2141  AT       2319
2142  WRITE    *AFTER WHICH ITEM DO YOU WANT
2143             TO INSERT THE NEW ITEM */
2144  JUMP     J3
2145  *
                                                                    MISS
2146  UNIT     MISS
2147  INHIBIT  ERASE
2148  FIGURE   2115,2141,2241,2215,2115
2149  JUMP     J22
2150  *
                                                                    J3
2151  UNIT     J3
2152  INHIBIT  ERASE
2153  CALC     V50←1
2154  ARROW    2629
2155  LONG     1
2156  ANSWER
2157  JUMP     J3
2158  MATCH    V2,1,2,3,4
2159  JUMP     V2,INS2,X
2160  JUMP     V(41+V2),NO;X
2161  CALC     V50←-1
2162  JUMP     V(31+V2),X,NO
2163  CALC     V(41+V2)←-1
2164  JUMP     V2,INS2,INS1
2165  *
                                                                    DEL2
2166  UNIT     DEL2
2167  INHIBIT  ERASE
2168  DELAY    .3
2169  AT       2629
2170  ERASE    3
2171  AT       2619
2172  WRITE    (1 TO 5)
2173  JUMP     J4
2174  *
                                                                    NO
2175  UNIT     NO
2176  INHIBIT  ERASE
2177  AT       2316
2178  ERASE    80
2179  AT       2616
```

```
ERASE    18
CALC     N1*V2+1
AT       2318
WRITEC   V50:      NODE ¬0AT,2321¬1 ¬0T,N1¬1 DOES NOT EXIST
         !    NODE ¬0AT,2319¬1 ¬0T,N1¬1 HAS BEEN DELETED
         ! A NEW NODE HAS BEEN INSERTED
           AFTER NODE ¬0AT,2423¬1 ¬0T,N1¬1, THERE IS NOT
WRITEC   V50:!!¬0AT,2520¬1 ENOUGH ROOM ON THE SCREEN
           TO INSERT ANOTHER NODE.
JOIN     PDP
AT       2318
ERASE    232
JUMP     J21
*
```

---------- INSERT
SPACE LEFT = 7
INS1

```
UNIT     INS1
LAB      AVAIL
INHIBIT  ERASE
CALC     V4*1
AT       2319
ERASE    80
AT       2616
ERASE    18
AT       2419
WRITE    *PRESS -*N*F*X*T- TO INSERT A NEW
              NODE AFTER ITEM
ADD1     V2
SHOW     V2,3
JOIN     NEARER
PAUSE
AT       2419
ERASE    80
JOIN     INS11
JUMP     J21
*
```

INS11

```
UNIT     INS11
MODE     ERASE
CALCC    V46¬=31+V2,V5*1606+12*V2,V5*1605+12*V2
LINE     1600+12*V2,V5
CALC     OR*1605+V2*12
JOIN     RARROW
GOTO     V46>40,X,EI4
LINE     1800+12*V2,1900+12*V2
LINE     1900+12*V2,1906+12*V2
```

EI4

```
ENTRY    EI4
MODE     WRITE
LINE     1800+V2*12,1600+V2*12
CALC     OR*1800+V2*12
JOIN     DARROW
CALC     OR*1799+V2*12
JOIN     NODE
CIRCLE   2,40+V2*96,208,15,360
```

```
2231  GOTO     V46>40,EI1,X
2232  GOTO     V46>31+V2,EI2,X
2233  LINE     1906+V2≠12,1706+V2≠12
2234  CALC     OR←1706+V2≠12
2235  JOIN     UARROW
2236  GOTO     EI3
                                                                    EI2
2237  ENTRY    EI2
2238  LINE     1906+V2≠12,1606+12≠V2
2239  GOTO     EI3
                                                                    EI1
2240  ENTRY    EI1
2241  GOTO     V46=45,EI5,X
2242  LINE     1906+V2≠12,1419+V46≠12
2243  CALC     OR←1419+V46≠12
2244  JOIN     RARROW
2245  GOTO     EI3
                                                                    EI5
2246  ENTRY    EI5
2247  MODE     ERASE
2248  LINE     1603+12≠V2,1803+12≠V2
2249  WRITE    ¬0AT,1802+12≠V2¬12KL
2250  MODE     WRITE
2251  DRAW     1906+12≠V2!1909+12≠V2!2109+12≠V2
2252  WRITE    ¬0AT,2108+12≠V2¬12KL
                                                                    EI3
2253  ENTRY    EI3
2254  AT       V2≠96-14,201
2255  WRITE    ITEM
2256  AT       30+V2≠96,201
2257  WRITE    A1≤/<V2
2258  AT       22+V2≠96,201
2259  SHOW     V2,1
2260  *
                                                                    INS2
2261  UNIT     INS2
2262  INHIBIT  ERASE
2263  DELAY    .3
2264  AT       2629
2265  ERASE    3
2266  AT       2619
2267  WRITE    (1 TO 4)
2268  JUMP     J3
2269  *
                                                                    DELETE
2270  UNIT     DELETE
2271  MODE     ERASE
2272  INHIBIT  ERASE
2273  FIGURE   2115,2141,2241,2215,2115
2274  JOIN     DELINS
2275  AT       2728
2276  ERASE    8
2277  JOIN     NEXT
2278  MODE     WRITE
2279  AT       2316
2280  WRITE    +WHICH ITEM DO YOU WANT TO DELETE +/
                                                                    J4
```

```
INHIBIT ERASE
CALC     V50⇒0
ARROW    2629
LONG     1
ANSWER
JUMP     J4
MATCH    V2,1,2,3,4,5
JUMP     V2,DEL2,X
JUMP     V(31+V2),X,NO
CALC     V(31+V2)⇒0
JUMP     V2,DEL2,DEL1
*
UNIT     DEL1                                                    DEL1
LAB      AVAIL
INHIBIT ERASE
CALC     V4⇒-1
JOIN     NEARER
JOIN     DEL11
JUMP     J21
*
UNIT     DEL11                                                   DEL11
INHIBIT ERASE
AT       2316
ERASE    34
AT       2616
ERASE    18
AT       165,121
ERASE    13
ADD1     V2
AT       2319
WRITE    +PRESS -+N+E+X+T- TO DELETE ITEM
SHOW     V2,2
PAUSE
AT       2319
ERASE    29
AT       96≠V2-62,249
ERASE    7
MODE     ERASE
CALC     OR⇒1593+12≠V2
JOIN     RARROW
CALC     OR⇒1493+12≠V2
JOIN     NODE
MODE     WRITE
GOTO     V46>40,DE1,X
GOTO     V36>40,DE2,X

------------------------------------------------------------------ MODIFY
                                                        SPACE LEFT = 5
LINE     1591+12≠V2,1229+12≠V46
GOTO     D24

ENTRY    DE1                                              DE1
GOTO     V46=45,DE3,X
GOTO     V36>40,DE4,X
```

```
2332  GOTO    V36=40,DE5,X
2333  MODE    ERASE
2334  LINE    1240+12*V36,1595+12*V2
2335  LINE    1600+12*V2,1600+12*V2
2336  CALC    OR=1800+12*V2
2337  JOIN    DARROW
2338  MODE    WRITE
2339  LINE    1240+12*V36,1540+12*V36
2340  LINE    1540+12*V36,1419+12*V46
2341  CALC    OR=1419+12*V46
2342  JOIN    RARROW
2343  GOTO    D24
                                                                    DE2
2344  ENTRY   DE2
2345  MODE    ERASE
2346  CALC    OR=1226+12*V36
2347  JOIN    UARROW
2348  MODE    WRITE
2349  LINE    1326+12*V36,1126+12*V36
2350  LINE    1594+12*V2,1229+12*V46
2351  GOTO    D24
                                                                    DE3
2352  ENTRY   DE3
2353  GOTO    V36>40,DE6,X
2354  JOIN    DEERA
2355  LINE    1243+12*V36,1593+12*V2
2356  MODE    WRITE
2357  LINE    1243+12*V36,1443+12*V36
2358  WRITE   ¬DAT,1442+12*V36¬12KL
2359  GOTO    D24
                                                                    DE6
2360  ENTRY   DE6
2361  MODE    ERASE
2362  JOIN    DEERA
2363  LINE    1126+12*V36,1592+12*V2
2364  LINE    1426+12*V36,1126+12*V36
2365  CALC    OR=1226+12*V36
2366  JOIN    UARROW
2367  MODE    WRITE
2368  FIGURE  1426+12*V36,1429+12*V36,1629+12*V36
2369  WRITE   ¬DAT,1628+12*V36¬12KL
2370  GOTO    D24
                                                                    DE5
2371  ENTRY   DE5
2372  MODE    ERASE
2373  CALC    OR=1800+12*V2
2374  JOIN    DARROW
2375  LINE    1600+12*V2,1900+12*V2
2376  LINE    1603,1601+12*V2
2377  MODE    WRITE
2378  LINE    1603,1903
2379  LINE    1903,1419+12*V46
2380  CALC    OR=1419+12*V46
2381  JOIN    RARROW
2382  GOTO    D24
                                                                    DE4
2383  ENTRY   DE4
```

```
LINE     1426+12*V36,1419+12*V46
CALC     OR*1419+12*V46
JOIN     RARROW
MODE     ERASE
LINE     1126+12*V36,1600+12*V2
LINE     1600+12*V2,1900+12*V2
LINE     1426+12*V36,1126+12*V36
CALC     OR*1226+12*V36
JOIN     DARROW
LINE     1120+12*V46,1320+12*V46
CALC     OR*1320+12*V46
JOIN     DARROW

ENTRY    D24                                                    D24
MODE     WRITE
*

UNIT     DEERA                                                  DEERA
MODE     ERASE
LINE     1600+12*V2,1603+12*V2
LINE     1603+12*V2,1803+12*V2
WRITE    ¬DAT,1802+12*V2¬12KL
*

UNIT     NEARER                                                 NEARER
INHIBIT  ERASE
CALCC    V4=1,V3*V2+1,V3*V2+1

ENTRY    N1                                                     N1
SUB1     V3
GOTO     V3=0$OR$V(30+V3)=-1,X*N1
CALC     V36*V3
CALCC    V4=1,V3*V2,V3*V2+1

ENTRY    N2                                                     N2
SUB1     V3
GOTO     V3=0$OR$V(40+V3)=-1,X*N2
CALCC    V3¬>V36,V36*40+V3,V36*30+V36
CALC     V3*V2

ENTRY    N3                                                     N3
ADD1     V3
GOTO     V3=6$OR$V(30+V3)=-1,X*N3
CALC     V46*V3
CALC     V3*V2

ENTRY    N4                                                     N4
ADD1     V3
GOTO     V3=5$OR$V(40+V3)=-1,X*N4
CALCC    V46¬<V3,V46*30+V46,V46*40+V3
*

UNIT     TERMD                                                  TERMD
TERM     D
PACK     N100,IDRILL
JUMP     YY
*
```

```
                                                                                                YY
2433  UNIT    YY
2434  ERASE
2435  JOIN    SCHAUS
                                                                                  INSISTE
2436  ENTRY   INSIST
2437  JUMPOUT STR2,IN2
2438  GOTO    INSISTE
2439  *
                                                                                                SCHAUS
2440  UNIT    SCHAUS
2441  AT      1612
2442  WRITE   ↑O↑N↑E ↑M↑O↑M↑E↑N↑T, ↑P↑L↑E↑A↑S↑E
2443  *
                                                                                                SANSFIN
2444  UNIT    SANSFIN
2445  TERM    0
2446  ERASE
2447  CALC    V1↑0
2448  JUMPOUT V1
2449  *

 ------------------------------------------------------------------------------------------------------------------------ CHARSP

 *** THIS IS A CHARSET   BLOCK ***

 ------------------------------------------------------------------------------------------------------------------------ CHARSP

 *** THIS IS A CHARSET   BLOCK ***

 ------------------------------------------------------------------------------------------------------------------------ CMO
                                                                                                SPACE LEFT = 37
                                                                                                LARROW
2450  UNIT    LARROW
2451  WRITE   ¬0AT,OR¬1≥C≥¬0AT,OR+100¬1≥D
2452  *
                                                                                                UARROW
2453  UNIT    UARROW
2454  WRITE   ¬0AT,OR+99¬1≥G≥¬0AT,OR+100¬1≥H
2455  *
                                                                                                RARROW
2456  UNIT    RARROW
2457  WRITE   ¬0AT,OR-1¬1≥A≥¬0AT,OR+99¬1≥B
2458  *
                                                                                                DARROW
2459  UNIT    DARROW
2460  WRITE   ¬0AT,OR-1¬1≥E≥¬0AT,OR¬1≥F
2461  *
                                                                                                NEXT
2462  UNIT    NEXT
2463  WRITE   ¬0AT,3022¬1↑PRESS -↑N↑E↑X↑T- TO LEAVE
2464  DRAW    2821;2844;3144;3121;2821
2465  WRITE   ¬0AT,169,32¬1↑PRESS -↑N↑E↑X↑T- TO LEAVE
2466  *
                                                                                                NODE
```

```
UNIT    NODE
DRAW    OR;OR+200;OR+208;OR+8;OR
*                                                                                FIG
UNIT    FIG
FIGURE  OR,OR+10,OR+310,OR+320,OR+20,OR+30,OR+330,OR+340,OR+40,OR,OR+300,OR+340
*                                                                                PRESS
UNIT    PRESS
WRITE   ¬0AT,3226¬1↑PRESS -↑N↑E↑X↑T-
*                                                                                PRESSEND
UNIT    PRESSEND
WRITE   ¬0AT,3252¬1↑PRESS -↑N↑E↑X↑T-
*                                                                                DEPRESS
UNIT    DEPRESS
AT      3226
ERASE   38
*                                                                                PDP
UNIT    PDP
JOIN    PRESS
JOIN    NEXT
JOIN    DEPRESS
*                                                                                WAIT
UNIT    WAIT
JOIN    PRESS
MODE    ERASE
ARROW   101
JKEY    BACK
SPECS   NOOKNO
LONG    2
OK
MODE    WRITE
ENDARROW
JOIN    KEY=NEXT$OR$KEY=BACK,DEPRESS,X
*                                                                                WAITN
UNIT    WAITN
MODE    ERASE
ARROW   101
SPECS   NOOKNO
LONG    2
OK
MODE    WRITE
ENDARROW
*                                                                                FROXES
UNIT    FROXES
CALC    OR+1505
JOIN    NODE
DELAY   .1
CALC    OR+1517
JOIN    NODE
DELAY   .1
```

```
2516  CALC    OR*1529
2517  JOIN    NODE
2518  DELAY   .1
2519  CALC    OR*1541
2520  JOIN    NODE
2521  DELAY   .1
2522  CALC    OR*1553
2523  JOIN    NODE
2524  LINE    1513,1517
2525  CALC    OR*1617
2526  JOIN    RARROW
2527  LINE    1625,1629
2528  CALC    OR*1629
2529  JOIN    RARROW
2530  LINE    1637,1641
2531  CALC    OR*1641
2532  JOIN    RARROW
2533  LINE    1649,1653
2534  CALC    OR*1653
2535  JOIN    RARROW
2536  *
```

DEPILE

```
2537  UNIT    DEPILE
2538  JOIN    FIGSTACK
2539  LINE    324,524
2540  CALC    OR*524
2541  JOIN    UARROW
2542  AT      321
2543  WRITE   *N*E*W*C*A*R*D
2544  AT      1134
2545  WRITE   *T*O*P
2546  LINE    1132,1128
2547  CALC    OR*1128
2548  JOIN    LARROW
2549  CIRCLE  2,163,336,10,360
2550  LINE    1119,1121
2551  CIRCLE  40,144,296,90,270
2552  CALC    OR*1620
2553  JOIN    RARROW
2554  CIRCLE  2,207,256
2555  LINE    1627,1629
2556  CIRCLE  40,224,216,90,-90
2557  CALC    OR*2128
2558  JOIN    LARROW
2559  CIRCLE  2,162,176,10,360
2560  LINE    2119,2121
2561  CIRCLE  40,144,136,90,270
2562  CALC    OR*2620
2563  JOIN    RARROW
2564  CIRCLE  2,207,96
2565  LINE    2627,2629
2566  CIRCLE  42,224,54,90,15
2567  WRITE   -0AT,2,33-12KL
2568  *
```

FIGSTACK

```
2569  UNIT    FIGSTACK
2570  CALC    OR*520
```

```
2571  JOIN    NODE
2572  CALC    OR*1020
2573  JOIN    NODE
2574  CALC    OR*1520
2575  JOIN    NODE
2576  CALC    OR*2020
2577  JOIN    NODE
2578  CALC    OR*2520
2579  JOIN    NODE
2580  *
2581  USE     CSCOMMENTS.COMENTCODE
```

UNIT	BLOCK	UNIT LOCATION	REFERENCES TO UNIT						
AD1	NEWCARD	511	446	509					
AL	ALLOCATION	1527	1544	1551	1553	1567	1570	1572	
ALGO	DISPLAY	413	446	486					
ALLOC	ALLOCATION	1531	184	1315	1529	1598			
ALLOC2	ALLOCATION	1581	1594						
AS	SEQUENTIAL	1630	1642	1648	1656	1661	1667		
AVA	INSERTION	1794	1718	1737	1741	1746	1751	1767	1776
			1785						
AVAIL	AVAIL LIST	1707	184	1679	1796	2002	2122	2195	2295
AVAIL2	INSERTION	1805	1708	1800					
AVB	INSERTION	1798	1817	1856	1875	1883	1891	1899	1909
			1927	1936	1949	1965	1973		
CADEF	MOVEDEQUE	1459	1106	1151	1358	1392			
CADEL	MOVEDEQUE	1452	1098	1144	1349	1387			
CARD	DECK	279	184	228	287	304	317	335	521
CHOICES	WELCOME	58	55	104					
CLEAN	DEQUE	1249	1096	1142	1347	1385			
CL1	DEQUE	1259	1256						
CL2	DEQUE	1263	1258						
COMEIN	NOT FOUND		13						
COMM	WELCOME	52	50	59					
COMMENTS	NOT FOUND		53						
CONCERN	CONTENTS	221	231	240					
DAR	POP-PUSH	933	891	918					
DARROW	CMO	2459	37	270	936	959	961	963	965
			1008	1011	1829	1835	1841	1847	1945
			2227	2337	2374	2395	2541		
DECK	DECK	332	281	349	360	421			
DEERA	MODIFY	2399	2354	2362					
DELETE	INSERT	2270	2130						
DELETEE	MOVEDEQUE	1448	1262	1330					
DELINS	DELETE	2087	2118	2136	2274				
DELRIGHT	MOVEDEQUE	1384	1344						
DEL1	INSERT	2294	2292						
DEL11	INSERT	2302	654	2299					
DEL12	OPERATIONS	651	649						
DEL2	DELETE	2166	2289	2292					
DEL-Q	MOVEQUEUE	1141	1055	1342					
DEM	MOVEQUEUE	1170	1163	1164					
DEPILE	CMO	2537	492						
DEPRESS	CMO	2479	616	2115	2486	2498			
DEQUE	MOVEQUEUE	1186	184	1053	1201	1216			
DESDEQUE	DEQUE	1214	1187	1303					
DESQUE	QUEUE	1001	973	1043					
DESTACK	STACK	808	770	860					
DE1	MODIFY	2329	2325						
DE2	MODIFY	2344	2326						
DE3	MODIFY	2352	2330						
DE4	MODIFY	2383	2331						
DE5	MODIFY	2371	2332						
DE6	MODIFY	2360	2353						
DG0	MOVEQUEUE	1159	1155						
DG1	MOVEQUEUE	1162	1158						
DG2	MOVEQUEUE	1179	1176						
DG3	MOVEQUEUE	1181	1178						

DISPA2	CONTENTS	174	172						
DISPLAY	DISPLAY	420	334	429	433	436	441	484	
DISPLAY2	DISPLAY	372	430	2035					
DOTDOT	QUEUE	1057	1034	1293					
DR0	MOVEDEQUE	1404	1398						
DT	QUEUE	1089	1093						
DU1	DEQUE	1302	1139	1184	1300	1317	1382	1415	1468
			1481	1484					
D24	MODIFY	2396	2328	2343	2351	2359	2370	2382	
EI1	INSERT	2240	2231						
EI2	INSERT	2237	2232						
EI3	INSERT	2253	2236	2239	2245				
EI4	INSERT	2223	2220						
EI5	INSERT	2246	2241						
EMPTY	MOVEDEQUE	1464	1143	1322	1386				
EM0	MOVEDEQUE	1477	1473						
EM2	MOVEDEQUE	1482	1478						
ERARROW	POP-PUSH	942	878	907					
ER0	POP-PUSH	894	890						
ES	DEQUE	1331	1339	1344					
EXP	INTRO	81	79	79	92	98	109		
FBOXES	CMO	2509	1005	1233	2058				
FIG	CMO	2470	256	325	341	344	347		
FIGDEQUE	DEQUE	1232	1218						
FIGSTACK	CMO	2509	817	841	1818	2538			
FRONT	MOVEDEQUE	1496	1035	1123	1156	1177			
FRONTT	MOVEDEQUE	1506	1499	1504					
FRONTT1	MOVEDEQUE	1501	1112	1123	1156	1177			
FULL	MOVEDEQUE	1470	1097	1321	1348				
IL0	MOVEDEQUE	1373	1367						
INFO	CONTENTS	227	188	233	238	247	250	254	260
			265	272	280				
INSERT	DELETE	2132	2130						
INSERTT	MOVEDEQUE	1444	1262	1330					
INSIST	SEQUENTIAL	1591	1589						
INSISTE	MODIFY	2436	2438						
INSLEFT	MOVEDEQUE	1346	1341						
INS1	INSERT	2194	2164						
INS11	INSERT	2214	602	2211					
INS2	INSERT	2261	2159	2164					
INS-Q	MOVEQUEUE	1095	1055	1343					
INTRO	CONTENTS	186	184	198	206	209	229		
J1	POP-PUSH	858	856	870	903	925			
J2	DELETE	2113							
J21	DELETE	2116	2192	2212	2300				
J22	DELETE	2120	2149						
J3	DELETE	2151	2144	2157	2268				
J4	INSERT	2281	2173	2287					
K	OPERATIONS	545	568	593	645	721			
LARROW	CMO	2450	461	970	1066	1130	1166	1235	1237
			1239	1241	1372	1400	2548	2558	
LA0	MOVEQUEUE	1134	1128						
LA1	MOVEQUEUE	1126	1122						
LA2	MOVEQUEUE	1114	1110						
LA3	MOVEQUEUE	1117	1113	1115					
LEFTF	MOVEDEQUE	1441	1127	1362	1378	1396			

UNIT	BLOCK	UNIT LOCATION	REFERENCES TO UNIT						
LEFTM	MOVEDEQUE	1434	1161	1180	1295	1362	1378		
LINKED	LINKED	2000	184	1589	1806	1998			
LINK1	LINKED	2057	561	588	640	669	716	2112	
LINK3	DELETE	2093	1709	2001					
LINK4	ALLOCATION	1511	1568	2018	2031				
LINK5	ALLOCATION	1514	1571	2021	2032				
LLIST	NEWCARD	519	184	483	533	554	574	771	1582
LP	LINKED	1996	2011	2017	2020	2022	2029	2034	2037
LUARROW	DEQUE	1278	1439	1509					
MISS	DELETE	2146	2130						
MORE	INTRO	127	108	140	176				
MV	QUEUE	1061	1071						
NEARER	MODIFY	2405	2207	2298					
NECA	NEWCARD	488	451	468	494	502			
NEG	POP-PUSH	885	880						
NEWCARD	NEWCARD	481	422						
NEXT	CMD	2462	1049	1257	1310	1326	2119	2139	2277
			2485						
NO	DELETE	2175	2160	2162	2290				
NODE	CMD	2467	379	385	391	677	684	691	940
			1879	2229	2323	2511	2514	2517	2520
			2523	2571	2573	2575	2577	2579	
NODEF	POP-PUSH	938	889	917					
NOGEN	OPERATIONS	623	608	632	660	728			
NOPDEL	OPERATIONS	657	649						
NOPER	OPERATIONS	605	570						
NOPINS	OPERATIONS	629	595						
NOPSPLIT	SPLIT	725	723						
NOTA-1	DISPLAY	396	434	443					
NOTA-2	DISPLAY	404	437	444					
NULL	POP-PUSH	920	908						
NUM	QUEUE	1076	1064	1074	1136	1173	1375	1408	
N1	MODIFY	2408	2410						
N2	MODIFY	2413	2415						
N3	MODIFY	2418	2420						
N4	MODIFY	2423	2425						
ONE	OPERATIONS	611	595						
OPCOMB	SPLIT	663	652	671	697	711	733		
OPDEL	OPERATIONS	635	598	642	665				
OPINS	OPERATIONS	583	573	590	636	653			
OPINS1	OPERATIONS	591	621	633					
OPLEFT	SPLIT	748	732	756					
OPPER	OPERATIONS	553	520	560	565	584	599		
OPPER1	OPERATIONS	566	609						
OPSPLIT	SPLIT	710	664	718	735	750			
OPSPLIT1	SPLIT	719	729						
OP1	OPERATIONS	572	570						
OP2	OPERATIONS	597	595						
OP4	SPLIT	731	723						
OR	MOVEDEQUE	1417	1268						
O1	OPERATIONS	643	661						
POP	CMD	2483	2189						
PO	POP-PUSH	901	884						
POP	POP-PUSH	876	870						
POPUSH	STACK	837	809	859					
PPDEQUE	DEQUE	1283	1215	1304					

PRNUEUE	QUEUE	1024	1002	1044					
PRESS	CMO	2473	439	614	2484	2489			
PRESSEND	CMO	2476	48	102	124	149	219	330	370
			479	543	581	603	655	708	746
			767	806	999	1022	1209	1230	1628
			1675	1705	1792	1994	2055		
PU	POP-PUSH	923	919						
PUSH	POP-PUSH	905	870						
QUEUE	QUEUE	972	184	868	987	1003	1188		
QUI	QUEUE	1042	1040	1055	1139	1184	1468	1476	
RARROW	CMO	2456	46	377	383	389	478	675	682
			689	845	931	1070	1133	1169	1369
			1403	1823	1871	1932	1961	2063	2219
			2244	2321	2342	2381	2386	2526	2529
			2532	2535	2553	2563			
REAR	MOVEDEQUE	1486	1036	1111	1137	1157			
RECTAN	MOVEQUEUE	1211	1063	1073	1135	1172	1374	1407	
RESET	DEQUE	1266	1183	1414					
RET	WELCOME	10	12						
RIGHTF	MOVEDEQUE	1430	1116	1160	1380				
RIGHTM	MOVEDEQUE	1420	1116	1137	1297	1397	1412		
RUARROW	MOVEDEQUE	1491	1489						
SANSFIN	MODIFY	2444	49	184					
SCHAUS	MODIFY	2440	2435						
SEQUEN	SEQUENTIAL	1596	184	1589	1607	1614	1636	1807	
SEQ2	SEQUENTIAL	1634	1597	1632	1680				
SEQ21	AVAIL LIST	1677	1635	1687	1694	1700			
SHOWN	POP-PUSH	948	855	902	924	1039	1138	1182	1311
			1381	1413					
SIDE	DEQUE	1319	1317						
STACK	STACK	769	184	749	784	810	974		
STARROW	POP-PUSH	953	819	846					
STATOP	POP-PUSH	967	818						
STEP	INSERTION	1802	1873	1881	1889	1897	1934	1947	1963
			1971						
STEP1	NEWCARD	448	500						
STEP2	NEWCARD	465	507						
TERM	INTRO	106	103	129					
TERMO	MODIFY	2428							
TITRE	WELCOME	16	14	56					
TOPAR	POP-PUSH	927	888	897	913	916			
TOPDO	POP-PUSH	872	898	911					
TRACK	QUEUE	1080	1033	1292					
UARROW	CMO	2453	34	579	1244	1247	1277	1281	1428
			1494	2235	2347	2366	2392		
WAIT	CMO	2488	54	91	97	139	197	205	208
			232	237	246	249	253	259	264
			271	286	303	316	348	359	428
			432	435	440	450	467	493	501
			532	559	564	589	641	670	696
			717	734	755	783	986	1200	1528
			1606	1613	1631	1686	1693	1699	1795
			1799	1997					
WAITN	CMO	2500	487	508	1105	1150	1357	1391	
WELCOME	WELCOME	1							
YY	MODIFY	2433	79	422	422	483	483	1315	1315

```
---------------------------------------------------------------------------------------------------------------------------------------- LOAD-1
                                                                                                                                SPACE LEFT = 2
                                                                                                                                   START2
 1  UNIT     START2
 2  WRITE    ¬0AT,1,22¬1(↑PLEASE STAND BY)                                                                                  RETR

 3  ENTRY    RETR
 4  CHARSET  STR1,CHARSP
 5  GOTO     ERROR=-1,RETR,X
 6  DEFINE   STUDENT
 7           QUES=N1
 8           OR=N3
 9           CHOICE=N17
10           LEVEL=N18
11           ERRORS=N20
12           OL=N24
13           TEST=N26
14           AR=N28
15           FXAR=N29
16           FYAR=N30
17           RES=N31
18           P=N32
19           S=N33
20           L=N34
21           M=N35
22           SCHATZ=N36
23           ANS=N38
24           B=N50
25           ARS=N65
26           CNT=N66
27           RESTE=N68
28           ANSVE=N69
29           REP=N70
30           OUT=N71
31           IN1=N100
32           IN2=N101
33           MYNAME=N145
34           SEGMENT,CHAR=ANS,12
35           SEGMENT,REM=REP,12
36  JOIN     MYNAME↑↑7STR2↑7,COMEIN
37  JUMP     IDRILL                                                                                                         QUES11

38  UNIT     QUES11
39  INHIBIT  ERASE
40  ********CHAR(1 ¬↑D 17) ↑N↑E↑X↑T[↑N↑E↑W↑C↑A↑R↑D]↑↑T↑O↑P
41  CALC     N38↑001160105013001240061
42           N39↑001160105012701030101
43           N40↑001220104006200650124
44           N41↑001170120777777777777
45           ANSVE↑-1
46           AR↑1635
47           L↑17
48  JUMP     EXPP
49  *                                                                                                                      QUES12
```

```
51  INHIBIT ERASE
52  ********CHAR(1 →↑D 11) → ↑T↑U↑P → ↑N↑E↑W↑C↑A↑R↑D
53  CALC     N38→00124011701200065011 6
54           N39→00105012701030101012 2
55           N40→00104777777777777777
56           ANSVE→
57           AR→2335
58           L→11
59  JUMP     EXPP
60  *                                                                    QUES211A

61  UNIT     QUES211A
62  INHIBIT ERASE
63  ******** CHAR(1 → 5) → ↑T→↑T↑1
64  CALC     N38→00124006501240045003 4
65           N39→07777000000000000000
66           ANSVE→1
67           AR→1229
68           L→5
69  JUMP     EXPP
70  *                                                                    QUES211B

71  UNIT     QUES211B
72  INHIBIT ERASE
73  ********CHAR(1 → 15) → ↑T>↑MTHENUNDERFLOW
74  CALC     N38→00124007301150024001 0
75           N39→00005001400170026000 5
76           N40→00022000400140017002 7
77           N41→07777000000000000000
78           ANSVE→2
79           AR→1429
80           L→15
81  JUMP     EXPP
82  *                                                                    QUES211C

83  UNIT     QUES211C
84  INHIBIT ERASE
85  ********CHAR(1 → 6) → ↑X[↑T]→↑Y
86  CALC     N38→00130006101240062006 5
87           N39→00131777700000000000
88           ANSVE→3
89           AR→1629
90           L→6
91  JUMP     EXPP
92  *                                                                    QUES212A

93  UNIT     QUES212A
94  INHIBIT ERASE
95  ********CHAR(1 → 16) → ↑T=0THENUNDERFLOW
96  CALC     N38→00124005400330024001 0
97           N39→00005001600250016000 4
98           N40→00005002200060014001 7
99           N41→00027777700000000000
--------------------------------------------------------------------------- LOAD-2
                                                                  SPACE LEFT = 2
100          ANSVE→4
```

```
101          AR←2329
102          L←16
103  JUMP    EXPP
104  *
                                                                    QUES212B
105  UNIT    QUES212B
106  INHIBIT ERASE
107  ********CHAR(1 ← 6) ← ↑Y←↑X[↑T]
108  CALC    N38←0013100650130006101 24
109          N39←0006277770000000000 00
110          ANSVE←5
111          AR←2529
112          L←6
113  JUMP    EXPP
114  *
                                                                    QUES212C
115  UNIT    QUES212C
116  INHIBIT ERASE
117  ********CHAR(1 ← 5) ← ↑T←↑T-1
118  CALC    N38←0012400650124004600 34
119          N39←0777700000000000000 00
120          ANSVE←6
121          AR←2729
122          L←5
123  JUMP    EXPP
124  *
                                                                    QUES221A
125  UNIT    QUES221A
126  INHIBIT ERASE
127  ********CHAR(1 ← 10) ← ↑R←↑MTHEN↑R←1
128  CALC    N38←0012200540115002400 10
129          N39←0000500160122006500 34
130          N40←0777777777777777777 77
131          ANSVE←7
132          AR←1029
133          L←10
134  JUMP    EXPP
135  *
                                                                    QUES221B
136  UNIT    QUES221B
137  INHIBIT ERASE
138  ********CHAR(1 ← 5) ← ↑R←↑R+1
139  CALC    N38←0012200650122004500 34
140          N39←0777777777777777777 77
141          ANSVE←8
142          AR←1246
143          L←5
144  JUMP    EXPP
145  *
                                                                    QUES221C
146  UNIT    QUES221C
147  INHIBIT ERASE
148  ********CHAR(1 ← 15) ← ↑R←↑FTHENOVERFLOW
149  CALC    N38←0012200540106002400 10
150          N39←0000500160017002600 05
151          N40←0002200060014001700 27
152          N41←0777777777777777777 77
```

```
153          ANSVE←9
154          AR←1429
155          L←15
156  JUMP    EXPP
157  *                                                                  QUES221D

158  UNIT    QUES221D
159  INHIBIT ERASE
160  ********CHAR(1 ⍴ 6) ⍴ ←X[←R]⍴←Y
161  CALC    N38←0013000610122006200065
162          N39←0013177777777777777777
163          ANSVE←10
164          AR←1629
165          L←6
166  JUMP    EXPP
167  *                                                                  QUES222A

168  UNIT    QUES222A
169  INHIBIT ERASE
170  ********CHAR(1 ⍴ 16) ⍴ ←R←←FTHENUNDERFLOW
171  CALC    N38←0012200540106002400 10
172          N39←0000500160025001600 04
173          N40←0000500220006001400 17
174          N41←0002777777777777777777
175          ANSVE←11
176          AR←2329
177          L←16
178  JUMP    EXPP
179  *                                                                  QUES222B

180  UNIT    QUES222B
181  INHIBIT ERASE
182  ********CHAR(1 ⍴ 10) ⍴ ←F←←MTHEN←F⍴1
183  CALC    N38←0010600540115002400 10
184          N39←0000500160106006500 34
185          N40←0777777777777777777777
186          ANSVE←12
187          AR←2529
188          L←10
189  JUMP    EXPP
190  *                                                                  QUES222C

191  UNIT    QUES222C
192  INHIBIT ERASE
193  ********CHAR(1 ⍴ 5) ⍴ ←F⍴←F+1
194  CALC    N38←0010600650106004500 34
195          N39←0777777777777777777777
196          ANSVE←13
197          AR←2745
198          L←5
199  JUMP    EXPP
200  *                                                                  QUES222D

201  UNIT    QUES222D
202  INHIBIT ERASE
203  ********CHAR(1 ⍴ 6) ⍴ ←Y⍴←X[←F]
204  CALC    N38←0013100650130006101 06
```

```
205          N39→00062777777777777777
206          ANSVE→14
207          AR→2929
208          L→6
209  JUMP    EXPP
210  *
                                                                                EXIT-G
211  UNIT    EXIT-G
212  INHIBIT ERASE
213  SHOW    ANSVE
214  JUMP    ANSVE<1,EXIT2,X
215  JUMP    ANSVE<7,EXIT4,EXIT5
216  *
                                                                                EXPP
217  UNIT    EXPP
218  INHIBIT ERASE
219  CALCC   ANSVE>0,N67→3122,N67→436
220  AT      N67
221  WRITE   -↑H↑E↑L↑P- IS AVAILABLE.
222  CALC    FXAR→800≠FRAC(AR/100)-8
223          RESTE→2≠(63-100≠FRAC(AR/100))
-------------------------------------------------------------------------------------------------------------------------------- QUESTION
                                                                                SPACE LEFT = 0
224          CNT→2
225          ARS→101
                                                                                EXP2
226  ENTRY   EXP2
227  CALC    FYAR→512-16≠INT(AR/100)
228          P→0
229          S→1
230          B→0
231  WRITE   ¬0AT,AR-1≥0
232  JOIN    DEPRESS
                                                                                RE1
233  ENTRY   RE1
234  INHIBIT ERASE
235  JOIN    ARROW
236  JUMP    (KEY=DATA1),EXIT-G,X
237  JUMP    (KEY=NEXT),JU1,X
238  JUMP    (KEY=DATA),IDRILL,X
239  JUMP    (KEY=ERASE),ERASE,X
240  JUMP    (KEY=ERASE1),ERASE1,X
241  JUMP    (KEY=HELP),H-D2,X
242  CALC    P→P+1
243          B→B+1
244          REM(P)→KEY
245  JOIN    SHO
246  GOTO    KEY=CHAR(B),RE1,X
247  GOTO    KEY=45,RB,X
248  AT      FXAR+8≠S,FYAR-16
249  WRITE   A=
250  JOIN    LOCATE
251  WRITE   NO
252  JOIN    ERRORS
                                                                                STUCK
253  ENTRY   STUCK
```

```
PAUSE
JUMP    (KEY=ERASE1)$OR$(KEY=NEXT),ERASE1,X
JUMP    (KEY=HELP),H-D2,X
GOTO    (KEY=ERASE),X,STUCK
AT      FXAR+8*S,FYAR-16
ERASE   1
JOIN    ERA1
GOTO    RE1
                                                                    RB
ENTRY   RB
CALC    B*H-1
GOTO    RE1
*
                                                                    JU1
UNIT    JU1
INHIBIT ERASE
JOIN    LOCATE
WRITEC  B=L,OK,NO
CALCC   B=L,ERRORS*ERRORS,ERRORS*ERRORS+1
JOIN    B=L,PRESS,X
                                                                    JU2
ENTRY   JU2
INHIBIT ERASE
JOIN    ARROW
JUMP    (KEY=HELP),H-D2,X
JUMP    (KEY=ERASE),ERASE,X
JUMP    (KEY=ERASE1),ERASE1,X
JUMP    (KEY=NEXT)$AND$B=L,RETURN,X
JUMP    (KEY=NEXT),ERASE1,JU2
*
                                                                    RETURN
UNIT    RETURN
INHIBIT ERASE
JOIN    B=L,DEPRESS,X
JUMP    (KEY=ERASE),ERASE,X
JUMP    (KEY=ERASE1),ERASE1,X
MODE    ERASE
CALCC   ANSVE>0,N67*3122,N67*336
AT      N67
WRITE   -*H*E*L*P- IS AVAILABLE.
MODE    WRITE
JUMP    ANSVE,INTER,D23,D41,D42,D43,D44,D45,RET4,D51,D52,D53,D54,D55,D56,D57,RET4
*
                                                                    ERASE
UNIT    ERASE
INHIBIT ERASE
JOIN    ERA1
JUMP    RE1
*
                                                                    ERASE1
UNIT    ERASE1
INHIBIT ERASE
JOIN    DEPRESS
AT      AR+1
ERASE   RESTE*INT(CNT/2)
JUMP    EXP2
*
```

```
                                                                                                                ERA1
UNIT    ERA1
JOIN    DEPRESS
AT      FXAR+8≠S,FYAR
ERASE   6
AT      FXAR+16,FYAR-16
ERASE   2
GOTO    S¬<2,ES1,X
CALC    S→S-1
GOTO    ES3
                                                                                                ES1
ENTRY   ES1
GOTO    FYAR=512-16≠INT(AR/100),ES2,X
CALC    FYAR→FYAR+16
        S→(504-FXAR)/8
GOTO    ES3
                                                                                                ES2
ENTRY   ES2
CALC    S→1
                                                                                                ES3
ENTRY   ES3
GOTO    P¬<0,EX1,X
GOTO    REM(P)=45,EX2,X
CALC    B→B-1
                                                                                                EX2
ENTRY   EX2
CALC    P→P-1
                                                                                                EX1
ENTRY   EX1
*
                                                                                                SHO
UNIT    SHO
INHIBIT ERASE
GOTO    S>(RESTE/2),X,SF1
CALC    S→2
        CNT→CNT+1
        FYAR→FYAR-16
GOTO    SF2
                                                                                                SF1
ENTRY   SF1
CALC    S→S+1
                                                                                                SF2
ENTRY   SF2
CALC    SCHATZ→KEY+06700
CALCC   KEY¬<26$OR$KEY=27$OR$KEY=28$OR$KEY=37$OR$KEY=38$OR$KEY=44$OR$KEY=49$OR$KEY=50$OR$KEY=53$OR$KEY=59,N37→FYAR-5,
        N37→FYAR
AT      FXAR+8≠S,N37
SHOWA   SCHATZ
*
                                                                                                LOCATE
UNIT    LOCATE
CALC    N67→FXAR+8≠S+24
GOTO    N67>496,X,JF1
AT      FXAR+16,FYAR-16
GOTO    JF2
                                                                                                JF1
ENTRY   JF1
```

```
AT       N67,FYAR                                                                        JF2

ENTRY    JF2
*

-------------------------------------------------------------------------------------------------------- COM-1
                                                                                          SPACE LEFT = 42
                                                                                             ERRORS
UNIT     ERRORS
CALC     ERRORS+ERRORS+1
*
                                                                                             RERRORS
UNIT     RERRORS
WRITE    +YOU MADE
SHOW     ERRORS,3
WRITEC   1-ERRORS, ERRORS    -+T+R+Y +T+H+I+S +D+R+I+L+L +A+G+A+I+N-, ERROR; ERROR    +THAT+7S PERFECT +.
*
                                                                                             NEXTLAB
UNIT     NEXTLAB
MODE     REWRITE
WRITE    -UAT,OR-1+PRESS -+L+A+B- TO TRY AGAIN.
         +PRESS -+N+E+X+T- TO QUIT.
DRAW     OR-101;OR-75;OR+125;OR+99;OR-101
MODE     WRITE
*
                                                                                             SKIP
UNIT     SKIP
WRITE    -0AT,307,407-1+PRESS -+D+A+T+A1- (SHIFT-DATA)  TO SKIP THIS SECTION.
DRAW     304,511;304,481;511,481;511,511;304,511
*
                                                                                             CARTE
UNIT     CARTE
DRAW     OR;OR+7;OR+207;OR+214;OR+14;OR+21;OR+221;OR+228;OR+28;OR;OR+200;OR+228
*
                                                                                             DESS
UNIT     DESS
CALC     OR+610
JOIN     CARTE
CALC     OR+1220
JOIN     CARTE
CALC     OR+1830
JOIN     CARTE
*
                                                                                             ARROW
UNIT     ARROW
MODE     ERASE
ARROW    101
JKEY     HELP,ERASE,ERASE1,DATA,DATA1
LONG     1
OK
ENDARROW
MODE     WRITE
*
                                                                                             DEPRESS1
UNIT     DEPRESS1
INHIBIT  ERASE
CALC     V5+WHERE
```

```
JOIN     DEPRESS
AT       V5
*                                                                          FUITE

UNIT     FUITE
AT       3120
JOIN     RERRORS
                                                                           EXIT1

ENTRY    EXIT1
NEXT     TEST,XX,DRILLB,XX                $$ NEWCARD(STR1)
BACK     TEST,XB,IDRILL,XB                $$ CARD(STR1)
LAB      SEP
DATA     IDRILL
CALC     OR←1903
JOIN     NEXTLAB
CALC     N100←↑7NEWCARD↑7
         N101←↑7CARD↑7
*                                                                          SEP

UNIT     SEP
CALC     ERRORS←0
         LEVEL←0
         OUI←0
MODPERM
JUMP     FILL
*                                                                          USE

UNIT     USE
FIGURE   N61-98,N61-93,N61+7,N61+2,N61-98
WRITE    ¬0AT,N61¬1    ↑U↑S↑E

         0,1,+,-
         >,[,],<,=
WRITEC   N60¬,¬0AT,N61+400¬1↑M,↑T,↑X,↑Y¬,¬0AT,N61+400¬1↑M,↑F,↑R,↑X,↑Y
WRITE    ¬0AT,N61+500¬1THEN
         OVERFLOW
         UNDERFLOW
FIGURE   N61-202,N61-189,N61+811,N61+798,N61-202
*                                                                          HELP1

UNIT     HELP1
INHIBIT  ERASE
BASE
JOIN     HELP
PAUSE
JOIN     EPB
GOTO     QEX
*                                                                          HELP2

UNIT     HELP2
JOIN     HELP
CALC     LEVEL←LEVEL+200
PAUSE
JOIN     EPB
*                                                                          EPB

UNIT     EPB
```

```
MODE    ERASE
JOIN    DESSIN
JOIN    NEWQUES
MODE    WRITE
*
                                                                    NEWQUES
UNIT    NEWQUES
WRITE   ¬0AT,117,16¬1↑PRESS ¬↑B↑A↑C↑K¬ TO GET A NEW QUESTION
FIGURE  3015,3050,3150,3115,3015
*
                                                                    HELP
UNIT    HELP
JOIN    ERRORS
MODE    ERASE
JOIN    PRESS
WRITE   ¬0AT,2312+LEVEL¬1↑PRESS ¬↑H↑E↑L↑P¬ OR,
MODE    REWRITE
CALC    N74←N23+LEVEL+4
WRITE   ¬0AT,N74¬1   ↑THE ANSWER IS
SHOW    N22,4
CALC    LEVEL←LEVEL-200
JOIN    NEWQUES
MODE    WRITE
JOIN    MONTRE
*
---------------------------------------------------------------------- DRILL=11
                                                                    SPACE LEFT = 66
                                                                    IDRILL
UNIT    IDRILL
BASE
ERASE
TERM    D
ERASE
AT      825
WRITE   1- DRILL ¬$ 1

        2- DRILL ¬$ 2

        3- DRILL ¬$ 3

        4- DRILL ¬$ 4

        5- DRILL ¬$ 5
WRITE   ¬0AT,2116¬1 ↑PRESS 1,2,3,4 OR 5 TO GO THROUGH
                ANY OF THESE SECTIONS.
ARROW   1832
WRITE   ¬0AT,140,112¬1↑YOU CAN RETURN TO THIS FRAME
        ANY TIME BY PRESSING THE ↑T↑E↑R↑M
        KEY, TYPING ↑,D↑, (DRILLS) AND
        PRESSING THE ↑N↑E↑X↑T KEY.
FIGURE  2418,2448,2848,2818,2418
AT      419
WRITE   ↑T↑Y↑P↑E 0 ↑I↑F ↑Y↑O↑U ↑W↑A↑N↑T ↑T↑O ↑Q↑U↑I↑T
FIGURE  318,345,445,418,318
LONG    1
MATCH   V1,0,1,2,3,4,5,6,7,8
```

```
497  JUDGE    V1,IGNORE,OK
498  CALC     N64*1215
499  JUMP     V1,X,SANSFINT,DRILLA,DRILLB,DRILLC,DRILLD,DRILLE,IDRILL,USE,SANSFINT
500  *                                                                          DRILLA

501  UNIT     DRILLA
502  TERM     1
503  JOIN     DEPRESS
504  CALC     OUI*0
505  SETPERM  12
506  JUMP     SEP
507  *                                                                          BRANCH

508  UNIT     BRANCH
509  INHIBIT  ERASE
510  JOIN     DEPRESS
511  RANDP    CHOICE
512  JUMP     OUI=4,FUITE,Q1
513  *                                                                          FILL

514  UNIT     FILL
515  NEXT     BRANCH
516  BACK     DRILLA
517  DATA1    EXIT1                $$ BYPASS BY PRESSING DATA1
518  INHIBIT  ERASE
519  JOIN     RANDOM
520  JOIN     PRESS
521  *                                                                          RANDOM

522  UNIT     RANDOM
523  RANDU    N5,2
524  RANDU    N6,2
525  RANDU    N7,2
526  RANDU    N8,4
527  RANDU    N9,4
528  RANDU    N10,4
529  RANDU    N11,13
530  RANDU    N12,13
531  RANDU    N13,13
532  RANDU    N14,999
533  RANDU    N15,999
534  RANDU    N16,999
535  RANDU    N21,2
536  MODE     REWRITE
537  JOIN     SKIP
538  WRITE    ¬CAT,212¬1↑T↑A↑G    ↑S↑U↑I↑T    ↑R↑A↑N↑K    ↑N↑E↑X↑T
539  WRITE    ¬0AT,302,424¬1↑T↑O↑P
540  FIGURE   650,750,742
541  CALC     OR*742
542  JOIN     LARROW
543  AT       402
544  WRITE    MEMORY ADDRESS
545  AT       605
546  SHOW     N14,3
547  AT       52,394
548  SHOW     N5-1,7
549  SHOW     N8,7
```

```
SHOW     N15,7
CALC     OR→610
JOIN     CARTE
AT       1012
WRITE    MEMORY ADDRESS
AT       1215
SHOW     N15,3
AT       132,294
SHOW     N6-1,7
SHOW     N9,7
SHOW     N12,7
SHOW     N16,7
CALC     OR→1220
JOIN     CARTE
AT       1622
WRITE    MEMORY ADDRESS
AT       1825
SHOW     N16,3
AT       212,202
SHOW     N7-1,7
SHOW     N10,7
SHOW     N13,7
WRITE    ¬0AT,422,202-1≥J1
CALC     OR→1830
JOIN     CARTE
*
```

---------- DRILL-12

SPACE LEFT = 27

Q1

```
UNIT     Q1
BASE
NEXT     BRANCH
HELP     HELP1
DATA1    EXIT1                  $$ BYPASS BY PRESSING DATA1
INHIBIT  ERASE
CALC     LEVEL→LEVEL+200
AT       2112+LEVEL
WRITEC   CHOICE,..↑T↑A↑G[↑T↑O↑P]=,↑T↑A↑G[↑N↑E↑X↑T[↑T↑O↑P]]=,↑T↑A↑G[↑N↑E↑X↑T[↑N↑E↑X↑T[↑T↑O↑P]]]=,↑S↑U↑I↑T[↑T↑O↑P]=,
         ↑S↑U↑I↑T[↑N↑E↑X↑T[↑T↑O↑P]]=,↑S↑U↑I↑T[↑N↑E↑X↑T[↑N↑E↑X↑T[↑T↑O↑P]]]=,↑R↑A↑N↑K[↑T↑O↑P]=,↑R↑A↑N↑K[↑N↑E↑X↑T[↑T↑O↑P]]=,
         ↑R↑A↑N↑K[↑N↑E↑X↑T[↑N↑E↑X↑T[↑T↑O↑P]]]=,↑N↑E↑X↑T[    ]=,↑N↑E↑X↑T[↑T↑O↑P]=,↑N↑E↑X↑T[↑N↑E↑X↑T[↑T↑O↑P]]=
GOTO     CHOICE,X,X,A,A,A,A,A,A,A,A,A,C,A,A
                                                                                                  C
ENTRY    C
AT       2117+LEVEL
SHOW     N(13+N21),3
                                                                                                  A
ENTRY    A
CALCC    CHOICE,N22→0,N22→0,N22→N5-1,N22→N6-1,N22→N7-1,N22→N8,N22→N9,N22→N10,N22→N11,N22→N12,N22→N13,N22→N(14+N21),
         N22→N15,N22→N16
CALCC    CHOICE,N22→0,N22→0,N23→2121,N23→2127,N23→2133,N23→2122,N23→2128,N23→2134,N23→2122,N23→2128,N23→2134,N23→2122,
         N23→2122,N23→2128
ARROW    N23+LEVEL
AT       N23+LEVEL+1
ERASE    50
ANSV     N22
```

```
601  CALC     OUI←OUI+1
602  REMOVE   CHOICE
603  JOIN     PRESS
604  NO
605  MODPERM
606  CALC     N62←2112+LEVEL
607           N63←N23+LEVEL
608           N64←N63+JCOUNT
609  WRITE    ¬0AT,N64+3¬1 NO
610  WRITE    ¬0AT,2312+LEVEL¬1↑PRESS -↑H↑E↑L↑P- OR,
611  JOIN     ERRORS
612  JOIN     PRESS
613  PAUSE
614  JOIN     KEY=HELP,HELP2,X
615  MODE     ERASE
616  WRITE    ¬0AT,N64¬1     NO
617  WRITE    ¬0AT,2312+LEVEL¬1↑PRESS -↑H↑E↑L↑P- OR,
618  CALC     LEVEL←LEVEL-200
619  JOIN     PRESS                                                    QEX

620  ENTRY    QEX
621  JOIN     RANDOM
622  JUMP     BRANCH
623  *                                                                 MONTRE

624  UNIT     MONTRE
625  CALCC    CHOICE,N1←N1,N1←N1,N72←461,N72←1061,N72←1661,N72←461,N72←1061,N72←1661,N72←461,N72←1061,N72←1661,N72←461
626           N72←461,N72←1061
627  CALCC    CHOICE,N1←N1,N1←N1,N73←414,N73←1024,N73←1634,N73←421,N73←1031,N73←1641,N73←428,N73←1038,N73←1648,N73←435
628           N73←435,N73←1045
629  CALC     N75←800×FRAC((N74+22)/100)-8
630  CALC     N76←519-16×INT((N74+22)/100)
631  JOIN     DESSIN
632  *                                                                 DESSIN

633  UNIT     DESSIN
634  CIRCLE   2,N75,N76
635  DRAW     N75,N76;480,N76;N72;N73;N73+100;
636  CALC     OR←N73+100
637  JOIN     DARROW
638  *                                                                 DRILLB

639  UNIT     DRILLB
640  PACK     N100,LLIST
641  BACK     TEST,XX,DRILLA,XX              $$     LLIST(STR1)
642  DATA1    EXIT2              $$ BYPASS BY PRESSING DATA1
643  TERM     2
644  INHIBIT  ERASE
645  ERASE
646  JOIN     SKIP
647  DRAW     530;563;763;730;530
648  WRITE    ¬0AT,238,416¬1↑WHAT ARE THE TWO STEPS TO INSERT
649           A NEW ITEM IN A PILE ↑/
650  WRITE    ¬0AT,1335¬1↑ASSUME ↑N↑E↑W↑C↑A↑R↑D IS POINTING
651           AT THE NEW ITEM AND ↑T↑O↑P IS
652           POINTING AT THE TOP OF THE
653           PILE.
```

```
655  GOTO     KEY=BACK,DRILLB,X
656  GOTO     KEY=DATA1,EXIT2,X
657  JOIN     DEPILE
658  JOIN     WAIT
659  GOTO     KEY=BACK,DRILLB,X
660  AT       1335
661  ERASE    95
662  JUMP     D21
663  *
```

DRILL-2

SPACE LEFT = 13

D21

```
664  UNIT     D21
665  BASE
666  INHIBIT ERASE
667  FIGURE   1002,1013,1913,1902,1002
668  DELAY    .2
669  FIGURE   1105,1110,1210,1205,1105
670  AT       1203
671  WRITE       ↑USE↑!
672
673           ↑≠<↑≡ *
674           ↑≠<↑≡ (
675           ↑≠<↑≡ )
676           ↑≠<↑≡ ↑T↑O↑P
677           ↑≠<↑≡ ↑N↑E↑X↑T
678           ↑≠<↑≡ ↑N↑E↑W↑C↑A↑R↑D
679  FIGURE   1334,1349,1449,1434,1334
680  AT       1435
681  WRITE    ↑TYPE STEP ONE
682  WRITE    ¬0AT,1535¬120
683  WRITE    ¬0AT,435¬1 =↑H↑E↑L↑P= IS AVAILABLE.
684  MODE     ERASE
685  JOIN     JD21
686  MODE     WRITE
687  JUMP     QUES11
688  *
```

INTER

```
689  UNIT     INTER
690  INHIBIT ERASE
691  AT       529
692  ERASE    107
693  WRITE    ¬0AT,3225¬1 ↑PRESS =↑N↑E↑X↑T= TO DO STEP ONE.
694  JOIN     WAIT
695  GOTO     KEY=BACK,DRILLB,X
696  AT       1128
697  ERASE    10
698  AT       1228
699  ERASE    2
700  LINE     1028,1228
701  CIRCLE   2,207,416
702  LINE     627,629
703  CIRCLE   40,224,376,90,-90
704  CALC     DR+1125
705  JOIN     LARROW
```

```
706  AT       936
707  WRITE    ↑T↑O↑P
708  FIGURE   2034,2049,2149,2134,2034
709  AT       2135
710  WRITE    ↑TYPE STEP TWO.
711  WRITE    ¬AT,2335¬120
712  MODE     ERASE
713  JOIN     JD22
714  MODE     WRITE
715  JUMP     QUES12
716  *
                                                                          D23
717  UNIT     D23
718  WRITE    ¬(AT,3225¬1 ↑PRESS ¬↑N↑E↑X↑T¬ TO DO STEP TWO.
719  JOIN     WAIT
720  GOTO     KEY=BACK,DRILLB,X
721  AT       934
722  ERASE    6
723  AT       219
724  ERASE    160
725  LINE     520,528
726  AT       610
727  WRITE    ↑T↑O↑P
728  LINE     614,620
729  CALC     OR←620
730  JOIN     RARROW
                                                                      EXIT2
731  ENTRY    EXIT2
732  INHIBIT  ERASE
733  NEXT     TEST,XX,DRILLC,XX              $$    LLIST(STR1)
734  BACK     TEST,XB,DRILLA,XB              $$    INTRO(STR1)
735  CALC     N100←↑7LLIST↑7
736           N101←↑7INTRO↑7
737  LAB      DRILLB
738  DATA     IDRILL
739  MODE     WRITE
740  CALC     OR←217
741  JOIN     NEXTLAB
742  *
                                                                          JD21
743  UNIT     JD21
744  AT       1836
745  WRITE    ↑THE ↑N↑E↑W ↑C↑A↑R↑D HAVE TO BE
746           LINKED TO THE ↑T↑O↑P OF THE
747           PILE, SO THE FIRST STEP
748           IS THE FOLLOWING↑:
749
750             ↑N↑E↑X↑T[↑N↑E↑W↑C↑A↑R↑D] ← ↑T↑O↑P.
751
752           (↑PRESS ¬↑B↑A↑C↑K¬ TO ANSWER)
753  *
                                                                          JD22
754  UNIT     JD22
755  AT       2536
756  WRITE    ↑T↑O↑P HAVE TO POINT TO THE TOP
757           OF THE NEW PILE, SO STEP TWO
758           IS THE FOLLOWNG↑:
```

```
          +T+O+P + +N+E+W+C+A+R+D.

        / (+PRESS -+B+A+C+K- TO ANSWER)
*                                                                                H-D2

UNIT     H-D2
INHIBIT  ERASE
MODE     WRITE
JOIN     ERRORS
JOIN     ANSVE,JD21,JD22,H-D211A                                                 L1

ENTRY    L1
PAUSE
GOTO     (KEY=BACK),X,L1
MODE     ERASE
AT       AR+1
ERASE    RESTE≠INT(CNT/2)
JOIN     ANSVE,JD21,JD22,H-D211A
MODE     WRITE
JUMP     EXPP
*                                                                                ASSUM

UNIT     ASSUM
AT       OR
WRITE       SEQUENTIAL ALLOCATION+!

         +ASSUMING THAT +,UNDERFLOW+, AND
         +,OVERFLOW+, CAN OCCUR WHAT ARE
         THE DIFFERENT STEPS TO+!
*                                                                                DRILL-31
-------------------------------------------------------------------------------------------
                                                                                SPACE LEFT = 19
                                                                                DRILLC

UNIT     DRILLC
PACK     N100,DESDEQUE
BACK     TEST,XX,X,XX                    $$ DESDEQUE(STR1)
DATA1    EXIT3           $$ BYPASS BY PRESSING DATA1
TERM     3
INHIBIT  ERASE
ERASE
JOIN     HEAD
JOIN     WAIT
GOTO     KEY=BACK,DRILLC,X
GOTO     KEY=DATA1,EXIT3,X                                                      RETRY3

ENTRY    RETRY3
ERASE
JOIN     SKIP
CALC     ERRORS+0
         N73+425
         V2+0
MODE     WRITE
WRITE    -DAT,106-1 STACK,DEQUE,QUEUE,YES OR NO
DRAW     44,510;266,510;266,494;44,494;44,510
SETPERM  10
```

```
                                                                                ICI
ENTRY    ICI
INHIBIT  ERASE
RANDP    QUES
CALC     V2→V2+1
         N75→N73-116
JOIN     DEPRESS1
RANDP    QUES
JUMP     V2=6,SALIR,A00
*                                                                               SALIR
UNIT     SALIR
MODE     WRITE
FIGURE   2919,2944,3044,3019,2919
AT       148,32
WRITE    ↑YOU HAVE MADE
SHOW     ERRORS,3
WRITEC   ERRORS-1,   ERROR,   ERROR,  ERRORS    ↑THAT↑7S TOO MANY
JOIN     WAIT
                                                                                EXIT3
ENTRY    EXIT3
CALC     N100→↑7ALLOC↑7
         N101→↑7LLIST↑7
NEXT     TEST,XX,DRILLD,XX              $$   ALLOC(STR1)
BACK     TEST,XB,DRILLB,XB              $$   LLIST(STR1)
LAB      RETRY3
DATA     IDRILL
INHIBIT  ERASE
AT       101
ERASE    260
MODE     WRITE
CALC     OR→321
JOIN     NEXTLAB
*                                                                               PDP1
UNIT     PDP1
INHIBIT  ERASE
CALC     V5→WHERE
JOIN     PDP
AT       V5
*                                                                               PRESS1
UNIT     PRESS1
INHIBIT  ERASE
CALC     V5→WHERE
JOIN     PRESS
AT       V5
*                                                                               SHOW
UNIT     SHOW
AT       N75
SHOW     V2,2
*                                                                               A1
UNIT     A1
ANSWER   <A> (STACK,STACKS)
```

```
858  UNIT     A2
859  ANSWER   YES
860  *                                                                    A3

861  UNIT     A3
862  ANSWER   <A> (DEQUE,DEQUES)
863  *                                                                    A4

864  UNIT     A4
865  ANSWER   NO
866  *                                                                    A5

867  UNIT     A5
868  ANSWER   <A> (QUEUE,QUEUES)
869  *
870  *                                                                    TEXT

871  UNIT     TEXT
872  AT       N74
873  WRITEC   N1¬,¬,
874           ¬,- *WHAT IS THE SIMPLEST KIND OF LINKED LIST*/
875           ¬,- *WITH A STACK THE *,YOUNGEST*, ITEM IS
876             ALWAYS REMOVED FIRST.
877           ¬,- *WHAT IS A PILE */
878           ¬,- *WHAT IS THE MORE GENERAL LIST AMONG
879             DEQUE, STACK AND QUEUE.
880           ¬,- *WITH A DEQUE THE *,OLDEST*, ITEM IS
881             ALWAYS REMOVED FIRST.
882           ¬,- *WHAT IS A *F*I*F*O LIST */
883           ¬,- *THE BOTTOM OF THE STACK IS THE LEAST
884             ACCESSIBLE ITEM.
885           ¬,- *WITH A QUEUE THE *,OLDEST*, ITEM IS
886             ALWAYS REMOVED FIRST.
887           ¬,- *WHAT IS A *L*I*F*O LIST */
888           ¬,- *ITEMS ARE INSERTED AT THE FRONT OF THE QUEUE.
889
890  *
```

DRILL-32

SPACE LEFT = 56

A00

```
891  UNIT     A00
892  DATA1    EXITS          $$ BYPASS BY PRESSING DATA1
893  INHIBIT  ERASE
894  JOIN     SHOW
895  CALC     N74*WHERE
896  JOIN     TEXT
897  CALC     N76*WHERE+112
898  ARROW    N76
899  JOIN     QUES,X,X,A1,A2,A1,A3,A4,A5,A2,A2,A1,A4
900  REMOVE   QUES
901  CALC     N73*WHERE
902  JOIN     PRESS
903  NO
904  CALC     N72*WHERE-112
905  WRITE    ¬DAT,N76*JCOUNT¬1      NO
```

```
906  JOIN     HELP3
907  JOIN     ERRORS
908  MODE     REWRITE
909  JOIN     NEWQUES
910  MODPERM                                                              AA

911  ENTRY    AA
912  PAUSE
913  JUMP     KEY=DATA1,EXIT3,X
914  GOTO     KEY¬=BACK,AA,X
915  MODE     ERASE
916  JOIN     NEWQUES
917  JOIN     SHOW
918  JOIN     TEXT
919  AT       N76
920  ERASE    40
921  JOIN     HELP3
922  CALC     V2←V2-1
923  MODE     WRITE
924  JUMP     ICI
925  *                                                                    HEAD

926  UNIT     HEAD
927  WRITE    ¬DAT,1 15¬1↑YOU WILL HAVE TO ANSWER SEVERAL QUESTIONS OR STATEMENTS.
928               ↑EACH TIME ANSWER ONE OF THE FOLLOWING WORDS↑!
929                    STACK,DEQUE,QUEUE,YES OR NO.
930  *                                                                    HELP3

931  UNIT     HELP3
932  AT       N72
933  WRITEC   N1¬,¬,
934           ¬,↑A STACK≤≤≤≤≤↑6↑6↑6↑6↑6 IS THE SIMPLEST KIND OF LINKED
935           SINCE DELETION AND INSERTION ARE BOTH
936           MADE AT THE SAME END OF THE LIST.
937           ¬,↑WITH A STACK WE ALWAYS REMOVE THE ITEM ON
938           TOP.I.E. THE ONE WHICH HAS BEEN INSERTED
939           MORE R≤↑6E≤↑6C≤↑6E≤↑6N≤↑6T≤↑6L≤↑6Y≤↑6 THAN ANY OTHER.
940           ¬,↑MANY PEOPLE USE THE WORD ↑,PILE↑, INSTEAD
941           OF S≤↑6T≤↑6A≤↑6C≤↑6K≤↑6.
942           ¬,↑A DEQUE≤≤≤≤≤↑6↑6↑6↑6↑6 IS MORE GENERAL THAN A STACK OR A
943           QUEUE SINCE BOTH DELETIONS AND INSERTIONS
944           CAN BE MADE AT THE TWO ENDS OF THE LIST.
945           ¬,↑WITH A DEQUE ONE CANNOT SAY THAT THE ITEM
946           REMOVED IS ALWAYS THE OLDEST OR THE YOUNGEST
947           SINCE INSERTIONS AND DELETIONS ARE MADE AT
948           THE ↑6S↑6≤W↑6≤O ENDS OF THE LIST.
949           ¬,↑Q≤↑6U≤↑6E≤↑6U≤↑6E≤↑6S≤↑6 ARE ALSO CALLED FIRST-IN FIRST-OUT
950           LIST (↑F↑I↑F↑O).
951           ¬,↑THE BOTTOM OF THE STACK IS THE LEAST ACCESSIBLE
952           ITEM, AND IT WILL NOT BE REMOVED UNTIL ALL
953           OTHERS ITEMS HAVE BEEN DELETED.↑THE TOP OF
954           THE STACK IS THE MOST ACCESSIBLE ITEM.
955           ¬,↑ITEMS ARE INSERTED AT THE REAR AND ARE ONLY
956           REMOVED WHEN THEY REACH THE FRONT POSITION.
957           SO THE OLDEST ITEM IS ALWAYS REMOVED FIRST.
958           ¬,↑STACKS ARE ALSO CALLED LAST-IN FIRST-OUT (↑L↑I↑F↑O).
959           ¬,↑ITEMS ARE INSERTED AT THE REAR AND DELETED AT
```

```
*
                                                                                                  DRILL-4
                                                                                                SPACE LEFT = 25
                                                                                                  DRILLD

UNIT     DRILLD
NEXT     D41
PACK     N100.SEQ2
DATA1    EXIT4              $$ BYPASS BY PRESSING DATA1
TERM     4
INHIBIT  ERASE
ERASE
CALC     OR*303
         ERRORS*0
JOIN     SKIP
JOIN     ASSUM
AT       1008
WRITE    *≠≤*≡ INSERT AN ITEM INTO A STACK.
JOIN     WAIT
GOTO     KEY=BACK,DRILLD,X
GOTO     KEY=DATA1,EXIT4,X
WRITE    ¬0AT,284,448¬1*T    *! STACK POINTER.
         *Y    *! ITEM TO INSERT.
         *X[*T]*! LOCATION OF THE STACK
         *M    *! MAXIMUM LENGTH OF THE
                  STACK.
FIGURE   336,364,864,836,336
CALC     N60*-1
         N61*2305
JOIN     USE
JOIN     WAIT
GOTO     KEY=BACK,DRILLD,X
GOTO     KEY=DATA1,EXIT4,X
AT       1214
WRITE    STEP ONE*!
WRITE    ¬0AT,1229¬1≥0
JUMP     QUES211A
*                                                                                                  D41

UNIT     D41
NEXT     D42
INHIBIT  ERASE
JOIN     DEPRESS
AT       1414
WRITE    STEP TWO*! IF
WRITE    ¬0AT,1429¬1≥0
JUMP     QUES211B
*                                                                                                  H-D211A

UNIT     H-D211A
AT       3122
ERASE    22
AT       AR+201
WRITEC   N69¬,¬,
         ¬, *T → *T+1
         ¬, *T >≤  *M  THEN OVERFLOW
```

```
1011          ¬, ↑X[↑T] ⇒ ↑Y
1012          ¬, ↑T=0  THEN UNDERFLOW
1013          ¬, ↑Y ⇒ ↑X[↑T]
1014          ¬, ↑T ⇒ ↑T-1
1015          ¬, ↑R=↑M  THEN  ↑R ⇒ 1
1016          ¬, ↑R ⇒ ↑R+1
1017          ¬, ↑R=↑F  THEN OVERFLOW
1018          ¬, ↑X[↑R] ⇒ ↑Y
1019          ¬, ↑R=↑F  THEN UNDERFLOW
1020          ¬, ↑F=↑M  THEN  ↑F ⇒ 1
1021          ¬, ↑F ⇒ ↑F+1
1022          ¬, ↑Y ⇒ ↑X[↑F]
1023  MODE    KEY=BACK,ERASE,REWRITE
1024  CALCC   ANSVF>12,N67⇒3220,N67⇒AR+402
1025  AT      N67
1026  WRITE   (↑DPRESS -↑B↑A↑C↑K- TO ANSWER)
1027  MODE    WRITE
1028  *                                                                    D42

1029  UNIT    D42
1030  NEXT    D43
1031  INHIBIT ERASE
1032  JOIN    DEPRESS
1033  AT      1614
1034  WRITE   STEP THREE↑!
1035  WRITE   ¬0AT,1629¬120
1036  JUMP    QUES211C
1037  *                                                                    D43

1038  UNIT    D43
1039  NEXT    D44
1040  INHIBIT ERASE
1041  JOIN    DEPRESS
1042  MODE    ERASE
1043  JOIN    USE
1044  MODE    WRITE
1045  AT      2008
1046  WRITE   ↑≠≤↑≡ DELETE AN ITEM FROM A STACK↑!
1047  AT      2314
1048  WRITE   STEP ONE↑! IF
1049  WRITE   ¬0AT,2329¬120
1050  JUMP    QUES212A
1051  *                                                                    D44

1052  UNIT    D44
1053  NEXT    D45
1054  INHIBIT ERASE
1055  JOIN    DEPRESS
1056  AT      2514
1057  WRITE   STEP TWO↑!
1058  WRITE   ¬0AT,2529¬120
1059  JUMP    QUES212B
1060  *                                                                    D45

1061  UNIT    D45
1062  INHIBIT ERASE
1063  JOIN    DEPRESS
```

```
AT       2714
WRITE    STEP THREE+!
WRITE    -0AT,2729-120
JUMP     QUES212C                                                        RET4

ENTRY    RET4
AT       3014
JOIN     REPRORS                                                         EXIT4

ENTRY    EXIT4
BASE
MODE     WRITE
PACK     N100,SEQUEN
NEXT     DRILLE
BACK     TEST,XX,DRILLC,XX          $$   SEQUEN(STR1)
LAB      DRILLD
DATA     IDRILL
CALC     OR+1738
JOIN     NEXTLAB
*
------------------------------------------------------------------------------------------------------------ DRILL=5
                                                                                          SPACE LEFT = 57
                                                                                            DRILLE

UNIT     DRILLE
PACK     N100,ALLOC
BACK     TEST,XX,DRILLD,XX           $$  ALLOC(STR1)
DATA1    EXITS           $$ BYPASS BY PRESSING DATA1
TERM     5
ERASE
INHIBIT  ERASE
CALC     ERRORS+0
JOIN     DEPRESS
CALC     OR+203
JOIN     SKIP
JOIN     ASSUM
AT       808
WRITE    +≠≤+= INSERT AN ITEM INTO A QUEUE+!
JOIN     WAIT
GOTO     KEY=BACK,DRILLE,X
GOTO     KEY=DATA1,EXITS,X
AT       340
WRITE    +F    +! FRONT POINTER.
         +R    +! REAR POINTER.
         +Y    +! ITEM TO INSERT.
         +X[+F]+! FRONT NODE.
         +N    +! NUMBER OF NODES.
FIGURE   233,263,763,739,239
CALC     N60+0
         N61+2315
JOIN     USE
JOIN     WAIT
GOTO     KEY=BACK,DRILLE,X
GOTO     KEY=DATA1,EXITS,X
AT       1014
WRITE    STEP ONE+!    IF
WRITE    -0AT,1029-120
```

```
JUMP     QUES221A
*
                                                                        D51
UNIT     D51
NEXT     D52
INHIBIT ERASE
JOIN     DEPRESS
AT       1236
WRITE    OTHERWISE
WRITE    ¬0AT,1246¬120
JUMP     QUES221B
*
                                                                        D52
UNIT     D52
NEXT     D53
INHIBIT ERASE
JOIN     DEPRESS
AT       1414
WRITE    STEP TWO↑!   IF
WRITE    ¬0AT,1429¬120
JUMP     QUES221C
*
                                                                        D53
UNIT     D53
NEXT     D54
INHIBIT ERASE
JOIN     DEPRESS
AT       1614
WRITE    STEP THREE↑!
WRITE    ¬0AT,1629¬120
JUMP     QUES221D
*
                                                                        D54
UNIT     D54
NEXT     D55
INHIBIT ERASE
JOIN     DEPRESS
MODE     ERASE
JOIN     USE
MODE     WRITE
AT       2008
WRITE    ↑#<↑≡ DELETE AN ITEM FROM A QUEUE↑!


                 STEP ONE↑!   IF
WRITE    ¬0AT,2329¬120
JUMP     QUES222A
*
                                                                        D55
UNIT     D55
NEXT     D56
INHIBIT ERASE
JOIN     DEPRESS
AT       2514
WRITE    STEP TWO↑!   IF
WRITE    ¬0AT,2529¬120
JUMP     QUES222B
```

```
*                                                                       D56

UNIT      D56
NEXT      D57
INHIBIT   ERASE
JOIN      DEPRESS
AT        2736
WRITE     OTHERWISE
WRITE     ¬0AT.2746¬1≥0
JUMP      QUES222C
*                                                                       D57

UNIT      D57
INHIBIT   ERASE
JOIN      DEPRESS
AT        2914
WRITE     STEP THREE+1
WRITE     ¬0AT.2929¬1≥0
JUMP      QUES222D                                          RETS

ENTRY     RETS
AT        3114
JOIN      RERRORS                                           EXITS

ENTRY     EXITS
PACK      N100.LINKED
NEXT      TEST,XX,IDRILL,XX              SS LINKED(STR1)
MODE      WRITE
BACK      DRILLD
LAB       DRILLE
CALC      OR→1939
JOIN      NEXTLAB
*
-------------------------------------------------------------------- CMO
                                                          SPACE LEFT = 41
                                                                   LARROW

UNIT      LARROW
WRITE     ¬0AT,OR-1≥C≥¬0AT,OR+100-1≥D
*                                                                  UARROW

UNIT      UARROW
WRITE     ¬0AT,OR+99¬1>G≥¬0AT,OR+100¬1≥H
*                                                                  RARROW

UNIT      RARROW
WRITE     ¬0AT,OR-1¬1≥A≥¬0AT,OR+99¬1≥B
*                                                                  DARROW

UNIT      DARROW
WRITE     ¬0AT,OR-1¬1≥E≥¬0AT,OR¬1≥F
*                                                                  NEXT

UNIT      NEXT
WRITE     ¬0AT.3022¬1+PRESS -+N+E+X+T- TO LEAVE
DRAW      2821;2844;3144;3121;2821
WRITE     ¬0AT.169.32¬1+PRESS -+N+E+X+T- TO LEAVE
```

```
*                                                                                                NODE
UNIT    NODE
DRAW    OR;OR+200;OR+208;OR+8;OR
*                                                                                                FIG
UNIT    FIG
FIGURE  OR,OR+10,OR+310,OR+320,OR+20,OR+30,OR+330,OR+340,OR+40,OR,OR+300,OR+340
*                                                                                                PRESS
UNIT    PRESS
WRITE   ¬0AT,3226¬1↑PRESS -↑N↑E↑X↑T-
*                                                                                                PRESSEND
UNIT    PRESSEND
WRITE   ¬0AT,3252¬1↑PRESS -↑N↑E↑X↑T-
*                                                                                                DEPRESS
UNIT    DEPRESS
AT      3226
ERASE   38
*                                                                                                PDP
UNIT    PDP
JOIN    PRESS
JOIN    NEXT
JOIN    DEPRESS
*                                                                                                WAIT
UNIT    WAIT
JOIN    PRESS
MODE    ERASE
ARROW   101
JKEY    BACK
SPECS   NOOKNO
LONG    2
OK
MODE    WRITE
ENDARROW
JOIN    KEY=NEXT$OR$KEY=BACK,DEPRESS,X
*                                                                                                WAITN
UNIT    WAITN
MODE    ERASE
ARROW   101
SPECS   NOOKNO
LONG    2
OK
MODE    WRITE
ENDARROW
*                                                                                                FBOXES
UNIT    FBOXES
CALC    OR+1505
JOIN    NODE
DELAY   .1
CALC    OR+1517
```

```
1261  DELAY   .1
1262  CALC    OR*1529
1263  JOIN    NODE
1264  DELAY   .1
1265  CALC    OR*1541
1266  JOIN    NODE
1267  DELAY   .1
1268  CALC    OR*1553
1269  JOIN    NODE
1270  LINE    1613,1617
1271  CALC    OR*1617
1272  JOIN    RARROW
1273  LINE    1625,1629
1274  CALC    OR*1629
1275  JOIN    RARROW
1276  LINE    1637,1641
1277  CALC    OR*1641
1278  JOIN    RARROW
1279  LINE    1649,1653
1280  CALC    OR*1653
1281  JOIN    RARROW
1282  *                                                          DEPILE

1283  UNIT    DEPILE
1284  JOIN    FIGSTACK
1285  LINE    324,524
1286  CALC    OR*524
1287  JOIN    DARROW
1288  AT      321
1289  WRITE   *N*E*W*C*A*R*D
1290  AT      1134
1291  WRITE   *T*O*P
1292  LINE    1132,1128
1293  CALC    OR*1128
1294  JOIN    LARROW
1295  CIRCLE  2,163,336,10,360
1296  LINE    1119,1121
1297  CIRCLE  40,144,296,90,270
1298  CALC    OR*1621
1299  JOIN    RARROW
1300  CIRCLE  2,207,256
1301  LINE    1627,1629
1302  CIRCLE  40,224,216,90,-90
1303  CALC    OR*2126
1304  JOIN    LARROW
1305  CIRCLE  2,162,176,10,360
1306  LINE    2119,2121
1307  CIRCLE  40,144,136,90,270
1308  CALC    OR*2620
1309  JOIN    RARROW
1310  CIRCLE  2,207,96
1311  LINE    2627,2629
1312  CIRCLE  42,224,54,90,15
1313  WRITE   -04T,2*33-12KL
1314  *                                                          FIGSTACK
```

```
UNIT     FIGSTACK
CALC     OR⇒520
JOIN     NODE
CALC     OR⇒1020
JOIN     NODE
CALC     OR⇒1520
JOIN     NODE
CALC     OR⇒2020
JOIN     NODE
CALC     OR⇒2520
JOIN     NODE
*
```

--- OUT
SPACE LEFT = 249

```
                                                                    TERMTERM
UNIT     TERMTERM
TERM     TERM
PACK     N100,TERM
JUMP     XX
*
                                                                    TERMM
UNIT     TERMM
TERM     M
PACK     N100,CHOICES
JUMP     XX
*
                                                                    TERMI
UNIT     TERMI
TERM     I
PACK     N100,DISPA
JUMP     XX
*
                                                                    XX
UNIT     XX
JOIN     SCHAU
JUMPOUT  STR1,IN1
GOTO     XX
*
                                                                    XB
UNIT     XB
JOIN     SCHAU
JUMPOUT  STR1,IN2
GOTO     XB
*
                                                                    SCHAU
UNIT     SCHAU
ERASE
AT       1612
WRITE    ↑O↑N↑E ↑M↑O↑M↑E↑N↑T, ↑P↑L↑E↑A↑S↑E
*
                                                                    SANSFINT
UNIT     SANSFINT
TERM     O
CALC     V1⇒0
JUMPOUT  V1
*
```

```
USE      CSCOMMENTS.COMENTCODE
*
```

UNIT	BLOCK	UNIT LOCATION	REFERENCES TO UNIT						
A	DRILL-12	592	588	588	588	588	588	588	588
			588	588	588	588			
AA	DRILL-32	911	914						
ARROW	COM-1	384	235	274					
ASSUM	DRILL-2	779	972	1093					
AOV	DRILL-32	891	815						
A1	DRILL-31	855	899	899	899				
A2	DRILL-31	858	899	899	899				
A3	DRILL-31	861	899						
A4	DRILL-31	864	899	899					
A5	DRILL-31	867	899						
BRANCH	DRILL-11	508	515	579	622				
C	DRILL-12	589	588						
CARTE	COM-1	373	378	380	382	553	564	575	
COMEIN	NOT FOUND		36						
DARROW	CMO	1205	637	1287					
DEPILE	CMO	1283	657						
DEPRESS	CMO	1225	232	283	300	306	396	503	510
			998	1032	1041	1055	1063	1090	1120
			1129	1138	1147	1162	1171	1179	1232
			1244						
DEPRESS1	COM-1	393	813						
DESS	COM-1	376							
DESSIN	DRILL-12	633	447	631					
DRILLA	DRILL-11	501	499	516	641	734			
DRILLB	DRILL-12	639	403	499	655	659	695	720	737
			829						
DRILLC	DRILL-31	787	499	733	796	1076			
DRILLD	DRILL-4	962	499	828	976	988	1077	1084	1191
DRILLE	DRILL-5	1082	499	1075	1097	1110	1192		
D21	DRILL-2	664	662						
D23	DRILL-2	717	291						
D41	DRILL-4	995	291	963					
D42	DRILL-4	1029	291	996					
D43	DRILL-4	1038	291	1030					
D44	DRILL-4	1052	291	1039					
D45	DRILL-4	1061	291	1053					
D51	DRILL-5	1117	291						
D52	DRILL-5	1126	291	1118					
D53	DRILL-5	1135	291	1127					
D54	DRILL-5	1144	291	1136					
D55	DRILL-5	1159	291	1145					
D56	DRILL-5	1168	291	1160					
D57	DRILL-5	1177	291	1169					
EPB	COM-1	445	436	443					
ERASE	QUESTION	293	239	276	284				
ERASE1	QUESTION	298	240	255	277	279	285		
ERA1	QUESTION	305	260	295					
ERRORS	COM-1	354	252	456	611	767	907		
ES1	QUESTION	314	311						
ES2	QUESTION	319	315						
ES3	QUESTION	321	313	318					
EXIT1	COM-1	402	517	581					
EXIT2	DRILL-2	731	214	642	656				
EXIT3	DRILL-31	825	790	797	892	913			
EXIT4	DRILL-4	1071	215	965	977	989			

UNIT	BLOCK	UNIT LOCATION	REFERENCES TO UNIT						
EXITS	DRILL-5	1187	215	1085	1098	1111			
EXIT-G	LOAD-2	211	236						
EXPP	LOAD-2	217	48	59	69	81	91	103	113
			123	134	144	156	166	178	189
			199	209	777				
EXP2	QUESTION	226	303						
EX1	QUESTION	327	322						
EX2	QUESTION	325	323						
FBOXES	CMO	1255							
FIG	CMO	1216							
FIGSTACK	CMO	1315	1284						
FILL	DRILL-11	514	417						
FOITE	COM-1	399	512						
HEAD	DRILL-32	926	794						
HELP	COM-1	455	434	440					
HELP1	COM-1	431	580						
HELP2	COM-1	439	614						
HELP3	DRILL-32	931	906	921					
H-U2	DRILL-2	764	241	256	275				
H-U211A	DRILL-4	1004	768	775					
ICI	DRILL-31	808	924						
IDRILL	DRILL-11	469	37	238	404	406	499	738	831
			1078	1189					
INTER	DRILL-2	689	291						
JU21	DRILL-2	743	685	768	775				
JU22	DRILL-2	754	713	768	775				
JF1	QUESTION	350	347						
JF2	QUESTION	352	349						
JU1	QUESTION	266	237						
JU2	QUESTION	272	279						
LARROW	CMO	1196	542	705	1294	1304			
LOCATE	QUESTION	345	250	268					
L1	DRILL-2	769	771						
MONTRE	DRILL-12	624	467						
NEWQUES	COM-1	451	448	465	909	916			
NEXT	CMO	1208	1231						
NEXTLAB	COM-1	362	408	741	837	1080	1194		
NODE	CMO	1213	1257	1260	1263	1266	1269	1317	1319
			1321	1323	1325				
POP	CMO	1229	842						
POP1	DRILL-31	839							
PRESS	CMO	1219	271	458	520	603	612	619	848
			902	1230	1235				
PRESSEND	CMO	1222							
PRESS1	DRILL-31	845							
QEX	DRILL-12	620	437						
QUES11	LOAD-1	38	687						
QUES12	LOAD-1	50	715						
QUES211A	LOAD-1	61	993						
QUES211B	LOAD-1	71	1002						
QUES211C	LOAD-1	83	1036						
QUES212A	LOAD-1	93	1050						
QUES212B	LOAD-2	105	1059						
QUES212C	LOAD-2	115	1067						
QUES221A	LOAD-2	125	1115						
QUES221B	LOAD-2	136	1124						

UNIT	BLOCK	UNIT LOCATION	REFERENCES TO UNIT						
QUES221C	LOAD-2	146	1133						
QUES221D	LOAD-2	158	1142						
QUES222A	LOAD-2	168	1157						
QUES222B	LOAD-2	180	1166						
QUES222C	LOAD-2	191	1175						
QUES222D	LOAD-2	201	1183						
Q1	DRILL-12	577	512						
RANDOM	DRILL-11	522	519	621					
RARROW	CMO	1202	730	1272	1275	1278	1281	1299	1309
RB	QUESTION	262	247						
RERRORS	COM-1	357	401	1070	1186				
RETR	LOAD-1	3	5						
RETRY3	DRILL-31	798	830						
RETURN	QUESTION	281	278						
RET4	DRILL-4	1068	291	291					
RET5	DRILL-5	1184							
REI	QUESTION	233	246	261	264	296			
SALIR	DRILL-31	817	815						
SANSFINT	OUT	1357	499	499					
SCHAU	OUT	1352	1343	1348					
SEP	COM-1	412	405	506					
SF1	QUESTION	336	331						
SF2	QUESTION	338	335						
SHO	QUESTION	329	245						
SHOW	DRILL-31	851	894	917					
SKIP	COM-1	369	537	646	800	971	1092		
START2	LOAD-1	1							
STUCK	QUESTION	253	257						
TERMI	OUT	1337							
TERMM	OUT	1332							
TERMTERM	OUT	1327							
TEAT	DRILL-31	871	896	918					
UARROW	CMO	1199							
USE	COM-1	419	499	986	1043	1108	1149		
WAIT	CMO	1234	654	658	694	719	795	824	975
			987	1096	1109				
WAITN	CMO	1246							
XB	OUT	1347	404	404	734	734	829	829	1350
XX	OUT	1342	403	403	641	641	733	733	789
			789	828	828	1076	1076	1084	1084
			1189	1189	1330	1335	1340	1345	

BIBLIOGRAPHIC DATA SHEET	1. Report No. UIUCDCS-R-74-615	2.	3. Recipient's Accession No.
4. Title and Subtitle A PLATO LESSON ON INFORMATION STRUCTURES			5. Report Date January, 1974 6.
7. Author(s) MICHEL JEAN CHESNEL			8. Performing Organization Rept. No.
9. Performing Organization Name and Address Department of Computer Science University of Illinois at Urbana-Champaign Urbana, Illinois 61801			10. Project/Task/Work Unit No. 11. Contract/Grant No.
12. Sponsoring Organization Name and Address Department of Computer Science University of Illinois at Urbana-Champaign Urbana, Illinois 61801			13. Type of Report & Period Covered 14.

15. Supplementary Notes

16. Abstracts

The purpose of the study was to write a lesson in the TUTOR language for the PLATO IV computer assisted instruction system of the University of Illinois.

17. Key Words and Document Analysis. 17a. Descriptors

CAI
Computer Assisted Instruction
Information Structures

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages
Release Unlimited	UNCLASSIFIED	123
	20. Security Class (This Page) UNCLASSIFIED	22. Price

FORM NTIS-35 (10-70) USCOMM-DC 40329-P71

FEB 19 1975

BOUND BY
HNM

UNIVERSITY OF ILLINOIS-URBANA
510.84 IL6R no. C002 no.613-617(1973
Internal report /
3 0112 088401036